Strategies and Technologies for **Un**ited and **R**esilient Critical **I**nfrastructures and Vital **S**ervices in Pandemic-Stricken **E**urope

D7.5 Infrastructure inspection tool and training guide V3

Document Identification			
Status	Final	**Due Date**	30/04/2025
Version	1.0	**Submission Date**	29/04/2025

Related WP	WP7	**Document Reference**	D7.5
Related Deliverable(s)	D7.1, D7.2, D7.3, D7.4	**Dissemination Level (*)**	PU
Lead Participant	XLB	**Lead Author**	Daniel Vladušič
Contributors	ATS, XLB, SKY, ICS, MZI, ELS, SZ, SZI, PIL, ACO, INS, HDE, TSA, TS, HER, CCL, INT	**Reviewers**	Radoš Čabarkapa, EKC George Tsakirakis, INT

Keywords:
Critical infrastructure, remote inspection, artificial intelligence, satellite imagery, UAV

(*) Dissemination level: **(PU)** Public, fully open, e.g. web (Deliverables flagged as public will be automatically published in CORDIS project's page). **(SEN)** Sensitive, limited under the conditions of the Grant Agreement. **(Classified EU-R)** EU RESTRICTED under the Commission Decision No2015/444. **(Classified EU-C)** EU CONFIDENTIAL under the Commission Decision No2015/444. **(Classified EU-S)** EU SECRET under the Commission Decision No2015/444.

Document Information

List of Contributors	
Name	**Partner**
Daniel Vladušič	XLB
Anja Zdovc	XLB
Luka Boljević	XLB
George Tsakirakis	INT
Milan Tarman	ICS
Tasos Gkamaris	SKY
Mario Triviño	ATS
Urban Bavčar	ELS
Josip Radman	MZI
Blaž Jemenšek	PIL
Tomaž Ramšak	SZ
Andrea Bello	HDE
José María Jiménez	ACO
Sean Travers	CCL

Document History			
Version	**Date**	**Change editors**	**Changes**
0.0	24/01/2025	Daniel Vladušič (XLB)	Skeleton of deliverable provided
0.1	24/02/2025	Daniel Vladušič (XLB)	Inputs to the general sections provided
0.2	05/03/2025	Urban Bavčar (ELS)	Contributions to chapter 5
0.3	12/03/2025	José María (ACO)	Contributions to chapter 5
0.7	21/03/2025	Daniel Vladušič (XLB)	Inputs merged from HDE, ATS, XLB, SZ
0.8	24/03/2025	Daniel Vladušič (XLB)	Contributions to chapters 1, 5 and 6
0.9	26/03/2025	Daniel Vladušič (XLB)	Applying corrections to references. Merge of input from XLB
0.95	31/03/2025	Daniel Vladušič (XLB)	Applying additions received by ATS and HDE.
0.96	01/04/2025	Daniel Vladušič (XLB)	Applying proofreading changes (CCL)
0.98	15/04/2025	Daniel Vladušič (XLB)	Final version for QA
0.99	16/04/2025	Juan Alonso (ATS)	Quality Assessment
1.0	29/04/2025	Aljosa Pasic (ATS)	Final version

Quality Control		
Role	**Who (Partner short name)**	**Approval Date**
Deliverable leader	Daniel Vladušič (XLB)	15/04/2025
Quality manager	Juan Alonso (ATS)	16/04/2025
Project Coordinator	Aljosa Pasic (ATS)	29/04/2025

Table of Contents

List of Tables

List of Figures

List of Acronyms

Abbreviation / acronym	Description
2FA	Two-Factor Authentication
AI	Artificial Intelligence
BVLOS	Beyond Visual Line Of Sight
CI	Critical Infrastructure
CNN	Convolutional Neural Network
DCNN	Deep Convolutional Neural Network
ESA	European Space Agency
EU	European Union
FN	False Negative
FP	False Positive
FPV	First Person View
FTPS	File Transfer Protocol Secure
GDPR	General Data Protection Regulation
GPT	Generative Pre-trained Transformer
GUI	Graphical User Interface
IoU	Intersection over Union
LEVIR	LEarning, VIsion and Remote
LLM	Large Language Model
LSI	Legacy Systems Integration
MAE	Mean Square Error
mAP	mean Average Precision
MLLM	Multimodal Large Language Model
MQTT	Message Queuing Telemetry Transport
MSBC	Multisource build-up change
MTOW	Maximum Take Off Weight
NeRF	Neural Radiance Fields
POI	Point of Interest
REST API	Representational State Transfer Application Programming Interface
SAM	Segment Anything Model
SAM-HQ	Segment Anything Model in High Quality
TP	True Positive
UAS	Unmanned Aircraft System
UAV	Unmanned Aerial Vehicle
UI	User Interface
VHM	Vegetation Height Model
VLM	Visual-Language Model
VLOS	Visual Line Of Sight
VQA	Visual Question Answering
WP	Work Package
YOLO	You Only Look Once

Executive Summary

The SUNRISE tool provides users with near-real-time information on the state of a critical infrastructure by analyzing image and video feeds. The analysis of the images and video feeds is performed through advanced AI methods and provides alerts based on the state of the infrastructure (e.g. degraded infrastructure raises an alert). This information includes details on damaged components, structural issues, corrosion, and obstructing vegetation. The tool uses data from satellite and UAV feeds and presents it in the user interface.

This document outlines the AI mechanisms developed to enhance the input data and present it in the SUNRISE Remote Inspection Tool, detailing the steps taken to integrate the satellite and UAV inspection tools, develop the dashboard UI and connect with legacy CI systems for improved functionality. It serves as a final technical report, describing most of the development paths taken. Therefore, the reader can view it as a standalone deliverable and will not need to revisit earlier deliverables.

This deliverable presents the state of the technologies developed within the SUNRISE project. It also presents a robust foundation upon which further development within CIs or other Consortia can build upon and address a wide range of inspection challenges.

D7.5 is the final iteration on implementing remote inspection tools for critical infrastructures. This deliverable is within the scope of WP7, and as such, builds on previous deliverables, including D7.1[45], D7.2[46] and D7.3[47]. Please note that D7.4[48] is omitted since it is solely focused on the description and piloting of Pilot 1. The implementation of Pilot 2 and the tools tested will be described in this deliverable.

This deliverable can also be treated as a standalone work regarding the methodology for remote infrastructure inspection and can be read without knowledge of previous deliverables. It provides the fulfillment levels of CI stakeholders' requirements in a qualitative way, that is, their actual understanding and assessment of the RII Tool and its components.

The deliverable precedes the final piloting phase (Pilot 2), which builds on the experiments in remote inspection methods used in Pilot 1. To this end, the tool itself is integrated and can be used during piloting. It consists of three main parts: the satellite imagery module, the UAV module and the GUI. The architecture of the tool remains stable and importantly, allows for easy installation locally at the CI premises. The reason for this is that the CIs seldom allow for any of their infrastructure to be hosted on public premises (i.e. on a public cloud). Therefore, it is very easy to move the tool to an infrastructure hosted within the CI premises.

Overall, the focus of this tool is to provide useful and impactful means to inspect typically vast and, in some cases, hard to reach infrastructures. Such capabilities are vital for ensuring the resilience and safety of critical infrastructures that serve society.

1 Introduction

In the context of monitoring critical infrastructure through satellites and UAVs, the data processing involves extracting valuable insights, detecting potential threats, and assessing the overall condition of the infrastructure. This processed information is then used to make informed decisions regarding maintenance, security measures and response strategies to mitigate risks and safeguard the critical assets. High-resolution images and videos of critical infrastructure are captured by satellite and UAV systems. Subsequently, raw data undergoes processing using AI tools to enhance imagery quality and extract pertinent information. The focus of the AI components, and thus our approach, is to reduce information/alert fatigue in operators and to reduce the need for human inspection, the latter being particularly important in cases of unforeseen temporary conditions where workforce availability may be restricted. This structured approach to data flow ensures that critical infrastructure operators have the necessary tools and insights to respond effectively to challenges and events within their operational environment.

The enhanced data is integrated into a user-friendly dashboard interface, empowering operators to visualize and analyze the information effectively. By offering operators a comprehensive view of the infrastructure, this process enables them to make well-informed decisions in response to various issues or events.

1.1 Purpose of the document

The purpose of this document is to present the final version of the Infrastructure Inspection Tool and Technical Training Guide, developed as part of Work Package 7 (WP7) within the SUNRISE project. This deliverable, D7.5, is the final in following series of deliverables D7.1[45], D7.2[46] and D7.3[47].

This deliverable refines and enhances the tools and methodologies for remote inspection of critical infrastructures. Please note that we skip the deliverable D7.4[48], which reports on the outcomes of testing in Pilot 1.

In this final version, we provide a comprehensive overview of the remote inspection module, which integrates satellite imaging, Unmanned Aerial Vehicle (UAV) imaging, and an interactive user interface. The document details the architecture, functionalities and deployment strategies of each component, demonstrating their applicability in real-world scenarios.

A key focus of this deliverable is also the preparation for the final validation of the integrated system through pilot trials conducted in collaboration with end-users. The first piloting trials have been instrumental in demonstrating the system's effectiveness in monitoring and assessing the condition of critical infrastructures, ensuring resilience and operational continuity. The first trials also offered the insight into the complex scenarios that each Critical Infrastructure (CI) operator must manage, before and during the remote infrastructure inspection with UAVs. The first trials also showed the limited applicability of satellite imagery in the use cases presented by these CI operators. Finally, with actual testing, CIs were given the insight into what is needed on their side, what personnel must be present, etc. This is also reported in Section 5 for each of the piloting operators.

Furthermore, this document serves as a technical training guide, offering practical insights and guidelines for stakeholders to effectively utilize the inspection tools. By using the advanced technologies such as artificial intelligence and computer vision, the tools facilitate proactive maintenance and risk mitigation, thereby enhancing the resilience of critical infrastructures across Europe.

In summary, D7.5 presents the final technical development of WP7, delivering a robust and user-centric solution for infrastructure inspection that addresses the evolving challenges in a pandemic-stricken Europe or any other unforeseen events.

1.2 Relation to other project work

WP7 deals with the development and implementation of the Remote Infrastructure Inspection (RII) tools and as such has significant interdependencies and interactions with other project activities within the SUNRISE project. We evaluated significant interdependencies between WP7 and WP2 which focused on strategy development and implementation. **In short, we evaluated, how remote infrastructure inspection tools, can contribute to strategic and operational thinking during unforeseen events, like pandemics.**

The tools developed in WP7 (Deliverables D7.[45]1, D7.2[46], D7.3[47], and report on their first testing in D7.4[48]) directly contribute to strategic resilience measures outlined in WP2 (D2.2). The findings, resulting from physical piloting and particularly, preparation for piloting, are fed into the final strategy, which is reported in D2.3. Particularly, these tools mitigate risks associated with personnel availability during pandemics or similar disruptions, thus enhancing operational resilience and continuity of operations in (CI) (i.e., business continuity). By employing satellite imagery and UAVs, these tools significantly reduce the reliance on personnel being physically present on-site, thereby addressing challenges posed by absenteeism and ensuring continuity of inspection activities during crisis conditions. This connection supports strategic goals related to operational efficiency, safety, and the sustainability of CI services.

Specifically, the remote inspection tools enable early detection and prioritization of infrastructure maintenance tasks, aligning closely with WP2's business continuity objectives. The strategic use of AI-driven anomaly detection, vegetation monitoring, and infrastructure change detection provides CI operators with predictive insights, facilitating timely maintenance decisions even under conditions of limited workforce availability. Furthermore, the training programs required for CIs, such as, e.g., ensuring its workforce is capable of operating UAV, directly impacts the strategic goal of maintaining critical services continuity.

Additionally, WP7's outputs, including satellite-based analysis and UAVs, feed into the broader strategic assessment performed in WP2, as they provide a practical validation of proposed strategies and ensure that operational realities are integrated into high-level strategic planning. This synergy helps to refine and validate the strategic frameworks developed in WP2, enhancing their relevance and applicability in real-world scenarios.

The link between WP7 and WP2 presents an integration point within the SUNRISE project, providing not only technological solutions but also strategic tools for robust decision-making and crisis preparedness in critical infrastructure management.

1.3 Changes from D7.3 [47] to D7.5

Table 1: Differences between D7.3 [47] and D7.5 summary.

Section in D7.5	Section in D7.3 [47]	Differences
Executive Summary	Executive Summary	Updated
1 Introduction	1 Introduction	Updated
1.1 Purpose of the document	1.1 Purpose of the document	Updated
1.2 Relation to other project work	1.2 Relation to other project work	Updated
1.3 The changes from D7.3 [47] to D7.5	1.3 The changes from D7.2[46] to D7.3[47]	Updated
1.4 Structure of the document	1.3 Structure of the document	Updated
	1.5 The description of end-users using the RII tool	Section deleted as piloting sections contain the needed information about personnel.

Section in D7.5	Section in D7.3 [47]	Differences
2 Satellite inspection tool	2 Satellite inspection tool	Updated, providing new developments and encompassing previous work in order to provide a unified view on work in SUNRISE
2.1 General context	2.1 General context	Updated, providing new developments and encompassing previous work in order to provide a unified view on work in SUNRISE
2.2 Architecture: high level design	2.2 Architecture: high level design	Unchanged
2.3 Tool modules description and lab validation	2.3 Tool modules description and lab validation	Updated, providing new developments and encompassing previous work in order to provide a unified view on work in SUNRISE
2.4 Applicability for the pilots	2.4 Applicability for the pilots	Updated, providing new developments and encompassing previous work in order to provide a unified view on work in SUNRISE
2.5 Deployment	2.5 Deployment	Minor changes
3 UAV inspection tool	3 UAV inspection tool	Updated
3.1 General context	3.1 General context	Minor changes
3.2 Architecture: high level design & use cases definition	3.2 Architecture: high level design	Minor changes
3.3 Tool modules description	3.3 Tool modules description	Minor changes
3.4 Tool modules lab validation	3.4 Tool modules lab validation	Minor changes
3.5 Deployment	3.5 Deployment	Minor changes
3.6 UAV platform lab integration	3.6 UAV platform lab integration	Updated to conform to the needs of the pilots.
4 User interface for remote infrastructure inspection	4 User interface for remote infrastructure inspection	Updated
5 Pilot trials execution	5 Pilot trials execution	Updated
5.1 Elektro-Slovenija, d.o.o. (ELES)	5.1 Elektro-Slovenija, d.o.o. (ELES)	Section, with updated evaluation and CI roles provided.
5.2 Ministry of infrastructure of Slovenia (MZI)	5.2 Ministry of infrastructure of Slovenia (MZI)	Section, with updated evaluation and CI roles provided.
5.3 Hydro Dolomiti Energia, s.r.l. (HDE)	5.3 Hydro Dolomiti Energia, s.r.l. (HDE)	Section, with updated evaluation and CI roles provided.
5.4 ACOSOL (ACO)		Section, with updated evaluation and CI roles provided.
	5.4 Designated areas	Section deleted, as this deliverable is primarily about technology.
	5.4.1 Piloting areas in Slovenia	Section deleted, as this deliverable is primarily about technology.
	5.4.2 Piloting area in Italy	Section deleted, as this deliverable is primarily about technology.

Section in D7.5	Section in D7.3 [47]	Differences
	5.4.3 Piloting area in Spain	Section deleted, as this deliverable is primarily about technology.
6 Conclusions	6 Conclusions	Updated
References	References	Updated
Annex I: Satellite inspection API User Guide	Annex I: Satellite inspection API User Guide	Updated
Annex II: UAV Image Processing API System User Guide	Annex II: UAV Image Processing API System User Guide	Updated

1.4 Structure of the document

This document is divided into six (6) primary chapters, including the current one, reflecting the purpose of the document, its framework within the project, and the structure of its contents.

All chapters are intended to be read without the need to revisit previous deliverables – D7.1 [45], D7.2 [46], D7.3 [47] and D7.4 [48]. Please note that we deem this deliverable a direct update of D7.3 and not D7.4. This distinction is based on the fact that deliverables D7.1 to D7.3 are technical in nature, addressing the development of the technology, while D7.4 specifically reports on the piloting activities carried out in Pilot 1. This means that Pilot 2 will be reported in the final reporting deliverable – D7.6, together with the results of the technology tested and described in this deliverable.

The following describes the other five chapters:

- **Chapter 2** provides updates on the high-level design of the satellite inspection tool's architecture for Infrastructure change detection and vegetation monitoring.
- **Chapter 3** specifies the work done for the UAV platform integration.
- **Chapter 4** discusses the user interface architecture for remote infrastructure inspection and details the process of integration and validation for the satellite component, UAV component and CI legacy systems.
- **Chapter 5** covers the execution of the Pilot 1 trials on CI areas, updating the content of D7.3 [47].
- **Chapter 6** summarizes the findings and outcomes of the work detailed in the previous chapters. It should be noted that even though each of the sub-modules has its own distinct chapter, satellite inspection, UAV inspection, and GUI - are all part of the same remote inspection module. As such, the integration of the tools at the end of the project must be complete.

We would like to reiterate that the chapters are meant to be read as an integrated whole, providing a comprehensive view of the work conducted. This is the final deliverable concerning the Remote Infrastructure Inspection Tools, developed in the SUNRISE project. As such, the reader can treat this document as the definitive reference on this topic within the project.

1.5 Glossary adopted in this document

Critical Infrastructure (CI): Power distribution networks, transportation networks, and information and communication systems are all examples of critical infrastructure. The defence of critical assets is indeed essential for ensuring the safety and well-being of the European Union (EU) and its citizens. The electrical grid, transportation systems, and information and communication networks are key examples of what is known as "Critical Infrastructures". These infrastructures are essential to maintain in order to ensure that vital societal functions continue to operate smoothly. Natural disasters, acts of terrorism, and criminal activities all have the potential to cause damage to or destroy essential infrastructure, which may have serious repercussions for both the safety of EU residents and the complete EU.
Critical Assets (CAs): Are the significant resources that support both the social and business parts of an economy. If some of these assets fail, it will bring significant issues for business continuity. This

does not mean that the likelihood of failing is high. For planning purposes, each business or organization must identify its critical assets and know the corresponding information about them.

Use Case – A description of the interaction between an actor (e.g. a user or system component) and the system, outlining the sequence of actions or steps taken to achieve a specific goal. It represents a technical or functional task relevant to a particular operational need.

Scenario - Scenarios provide narrative or technical framing for a use case, with two types:

- **Contextual Scenario:** A broader, real-world context such as a pandemic or multi-hazard threat environment that may influence multiple systems.
- **Use Case Scenario:** A focused variant or narrative path within a specific use case, detailing alternative technical or procedural flows.

Pilot: Real-world validation phase of the project, built on defined use cases and scenarios. It involves the deployment and evaluation of solutions under realistic conditions.

2 Satellite inspection tool

This section describes the Satellite inspection tools, which use satellite imagery accompanied by AI methods to inspect CI. We will first outline the context and rationale for the need of such a tool (Section 2.1), followed by a high level description of the components (Section 2.2). We will then describe the components, summarizing the work completed and providing details on the methods used in the final solution (Section 2.3). Lastly, we will describe the solution's deployment (Section 2.4) and the applicability of the tool in all the pilots (Section 2.5).

While all our work is summarized here, further details on the background of the Satellite inspection tool can be found in D7.1[45]. For a detailed description of the methods tested, refer to D7.2[46] and D7.3[47]. Information regarding the tool validation conducted during Pilot 1 is available in D7.4[48].

2.1 General context

One approach to remote inspection is the use of satellite imagery. The main advantage of this approach is the ability to monitor large areas in a continuous and non-invasive way. With a growing number of satellite providers, high resolution imagery can be collected every few days. Our satellite inspection tool can process optical and multispectral satellite imagery to detect overgrown vegetation and other changes in the area. This can support remote inspection of critical infrastructure by detecting events, which might pose danger to the CI. On the other hand, the main limitation in this approach is the resolution of the imagery, which restricts the level of detail and the scale of changes that can be detected. Therefore, this tool can also be used as a trigger for more localised and detailed UAV inspection (Section 3).

Our tool consists of two components: i) infrastructure change monitoring and ii) vegetation management.

One of the potential threats to critical infrastructure is overgrown vegetation. Our first component supports vegetation monitoring by estimating the height of vegetation from freely available multispectral satellite imagery (Section 2.3.2). Depending on the vicinity of the critical infrastructure, detected vegetation height can be used for threat estimation.

Other threats identified in D7.1 are environmental events such as landslides, leaks, and human imposed events such as illegal build-ups. It is crucial to detect such events in a timely manner in order to check for potential damages. Our second component provides a general solution for detecting changes in the vicinity of the CI based on optical satellite imagery (Section 2.3.1).

2.2 Architecture: High level design

Our tool consists of the main satellite inspection module and two sub-modules, as depicted in Figure 1. The main module handles data collection and pre/post-processing. The two sub-modules denoted as Change and Vegetation are responsible for Change Detection and Vegetation height estimation.

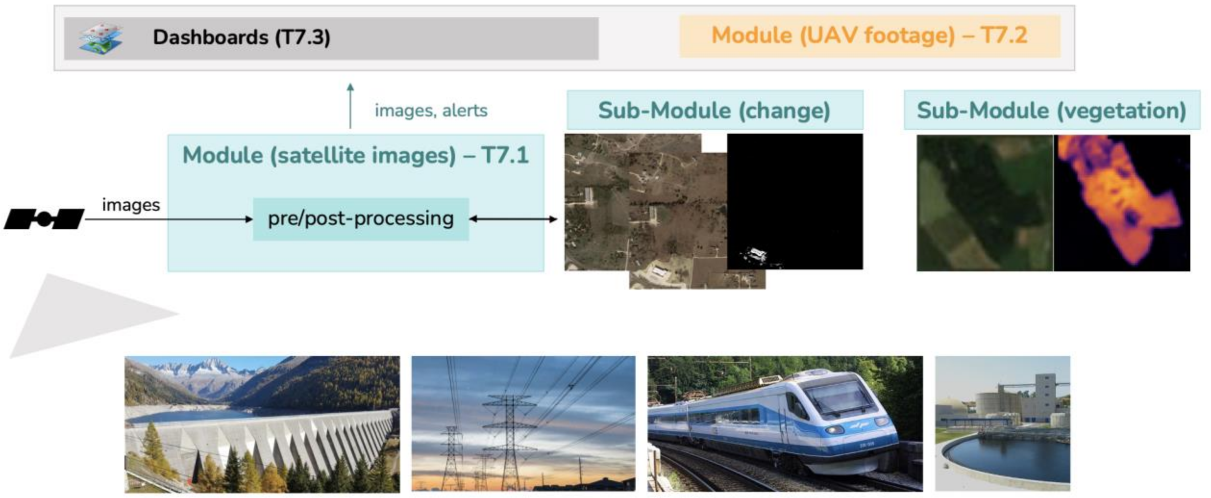

Figure 1: High-level architecture of the satellite-based inspection module and its sub-modules (T7.1)[45]

We provide our solution as an API with two separate endpoints for Vegetation Height and Change detection respectively. A request specifies the area and relevant dates, for which satellite imagery is downloaded and preprocessed. Our AI model then provides pixel-wise prediction returning vegetation height (in meters) or the probability of a change occurring. The Vegetation Height module is based on freely available imagery, while the Change Detection module requires a subscription with a commercial satellite imagery provider for the requested areas. An overview of the process is presented in the flowchart in Figure 2.

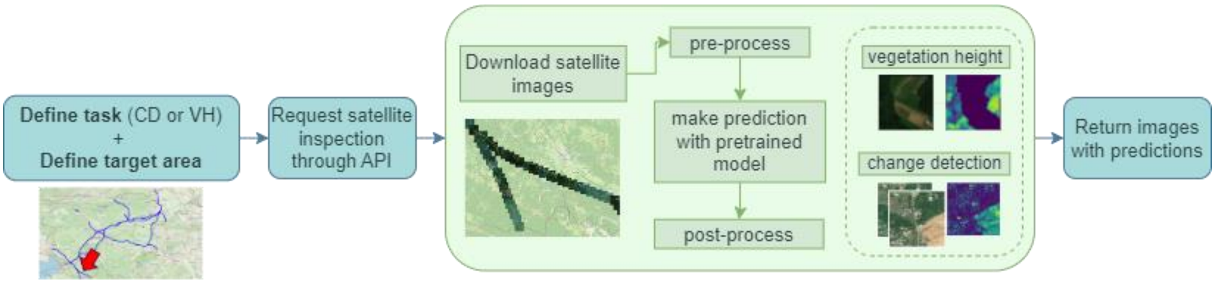

Figure 2: Satellite inspection tool request flowchart

2.3 Tool modules description

2.3.1 Infrastructure change detection

While detecting damage to the infrastructure is not feasible with satellite imagery due to limiting resolution, the inspection can still focus on detecting any bigger events or changes in the vicinity, which can pose threats to the CI. Such detection can then serve as a trigger for manual or UAV inspection. Risks can be environmental events such as landslides and floods, or human-made such as illegal constructions. Such larger-scale events can be identified with high resolution satellite imagery, which is the basis of our first submodule – change detection. An AI model is used to compare two satellite images taken at different points in time, detecting any unusual changes between them. While our model is designed for general-purpose change detection, our methods were validated on detection of landslides and construction.

One of the main limitations was the resolution of satellite imagery, which restricts the minimum size of events that can be reliably detected. We therefore evaluated multiple optical satellite imagery resolutions and providers. We tested and evaluated multiple methods, which we compared using publicly available change detection datasets. Here, we briefly summarize the methods and results, the details of which can be found in D7.2[46] and D7.3[47].

In the next sections, we present the unsupervised method we use in the end solution, the evaluation of different satellite imagery and the validation of the solution.

2.3.1.1 AI method

The change detection task in computer vision aims to identify any significant changes between two or more images. The two main paradigms are 1) supervised methods, which are based on training machine learning methods on labelled data, which usually focus on some type of change (e.g. new buildings) and 2) unsupervised methods, where no labels are needed and are therefore able to detect more general changes.

Supervised methods have two main disadvantages 1) they require extensive data labelling, which is expensive to acquire as it requires human labellers and 2) it lacks ability to detect novel events, since the datasets usually focus on specific changes. Additionally, satellite imagery is extremely variable, meaning that it is hard to transfer models trained on one type of imagery to another. We first experimented with supervised methods (ChangerEx [9] and Bitemporal Image Transformer[10]) but then switched to unsupervised methods due to the reasons described above. We tested two main methods: CDLR (Change Detection based on Image Loss Reconstruction)[13] and a method based on Latent Diffusion models (LDM)[14]. The LDM method achieved a higher mean F1 score and mean intersection over union (IoU). We therefore chose this method to use in our end solution.

Our method consists of two components:

1) Encoder backbone: Provide an input image, process it and extract important features
2) Cosine similarity head: Provide features of two input images and compute the pixel-wise cosine similarity. Pixels with a higher similarity score are labelled as no-change while low similarity score suggests a higher probability of change.

The feature extracting backbone we use is based on latent diffusion models (LDM) [14]. Diffusion models are mainly used for generating images from text by denoising a random image in multiple steps. Some methods such as Open-Vocabulary Panoptic Segmentation (ODISE) model [15] use diffusion for other tasks such as image segmentation. We follow the approach from [15] for generating features. Given the input image, the method first adds noise and then uses the pre-trained diffusion model to denoise the image.

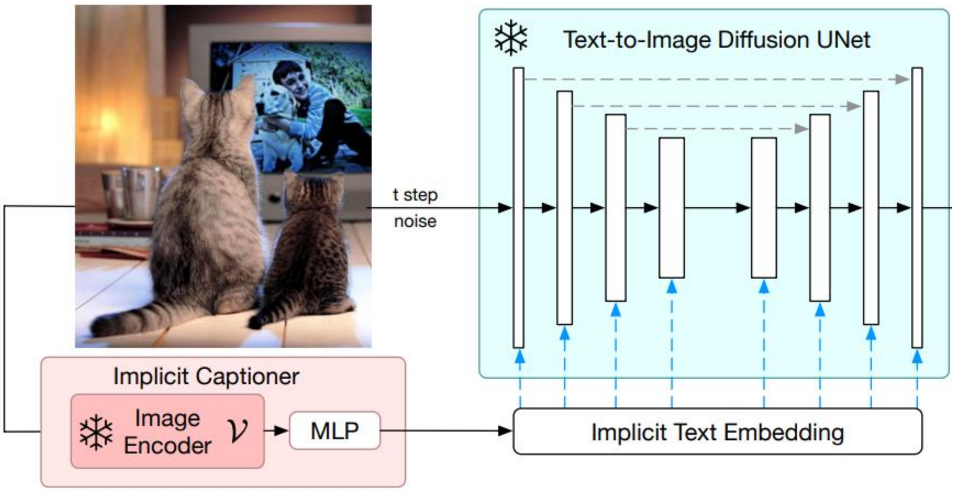

Figure 3: Image feature extractor of the ODISE model [15]

Comparison of different change detection methods requires a labelled dataset, where the prediction of the model is compared to the ground truth label. Our focus was the LEVIR-CD [8] dataset as it is the most widely used in literature while simultaneously corresponds to our use case of detecting new buildings. The dataset consists of pairs of before and after images captured by Google Earth with ground truth image labelling new buildings. We achieved a mean F1 score of 91.77% with the supervised ChangerEx model, 49.4% with CDLR and 53.6% with the LDM-CD method. Note that the high difference in score between supervised and unsupervised method is expected and we nevertheless focused on unsupervised methods for the reasons mentioned above. The evaluation showed that our method based on Latent Diffusion models (LDM-CD) is capable of detecting changes. In the next sections we describe how we validated the method with the pilots.

Possible future directions:

Additionally, when considering unsupervised methods, we briefly explored how to adapt the method to a specific use case (such as detecting buildings). For such an approach we would require a labelled dataset as mentioned above, which is difficult to obtain. We explored the possibility of generating the training dataset automatically. Tools such as Inpaint Anything[58] would enable us to artificially generate labelled datasets from the existing imagery by modifying it with generative models. We explored how to remove or add buildings from satellite imagery as seen in Figure 4.

Figure 4: SkySat optical satellite image (50cm resolution) with removed buildings. The left image shows the original, the middle is the mask of the houses to be removed, the right has houses removed with Inpaint Anything

Note that only very high-resolution imagery can be used for such dataset generation. We, however, evaluated that medium resolution imagery is the most suitable for our use case, as described in the next section. For this reason, we used the previously mentioned unsupervised method and left the possibility of fine tuning our method with the generated dataset as a possible future option.

2.3.1.2 Satellite imagery

Since our approach is based on unsupervised vision models, any optical satellite imagery can be used as input. Note that only RGB channels of the images are used. The main factor of determining the suitability of imagery is spatial resolution – that is the physical distance represented by a single pixel in the image. Resolution determines the scale of changes that can be observed. Other relevant factors in choosing a suitable satellite image provider are the frequency of the image capture over the same area, pricing and availability of imagery.

We have identified and evaluated 3 types of satellite images:

1. Sentinel 2: Sentinel-2 satellites are a part of the European Space Agency's (ESA) Copernicus Programme. It provides freely available images globally. It has a revisit time of 5 days and 10m spatial resolution. Such a low resolution is unfortunately **not suitable** for this use case.
2. PlanetScope: Imagery provided by the private company Planet that are captured by a constellation of small satellites. It has almost a daily revisit time and 3-4m spatial resolution. Bigger changes (such as big landslides and new construction) can be observed, while smaller changes cannot be observed due to lower resolution. This evaluation is presented in the next section. The left image in Figure 5 shows an example of a PlanetScope image. Planet offers a subscription plan, where the user is available to download all the available imagery over the given area[59]. The price is between 315 and 1110 EUR (depending on the area) for an area of 5sqkm. Despite its lower resolution, this option was identified as the most suitable for large-scale continuous inspection due to images being readily available for large areas.

3. SkySat imagery: Imagery provided by private company Planet. Satellites provide very high-resolution imagery (50cm). The right-hand image of Figure 5 shows an example. They are tasking satellites – the user requests an image to be captured within a 2-week window. As such, it has limitations such as pricing and the shape of tasking area (linear infrastructure such as powerlines or railways are not suitable since the area of interest is stretched out). This option was identified as being suitable for inspecting smaller, more targeted areas.

Ultimately, the most suitable option identified was PlanetScope imagery for large-scale and continuous inspection, which can be complemented by a more targeted and less frequent SkySat satellite inspection.

Figure 5: PlanetScope (left) and SkySat (right) satellite images of the same area

2.3.1.3 Tool validation

We validated our method at the target areas with the suitable satellite imagery. We required historical data on the events we wanted to detect paired with satellite imagery. For each event we required two images, one before and one after the event.

There were three sources of historical events data:

1. Illegal Buildup data from ELES: CI partner ELES provided a dataset of historical detection of illegal building activities along their electricity grid. Data was obtained by ELES personnel and includes 115 detected events between 2013 to 2023. This data includes the date of the event detected, the location and a short description. Note that the date of the detected event does not necessarily correspond to the date of the event occurring due to the low frequency of checks. Additionally, many events were too small to be observed, such as as the construction of a new fence. Finally, only events since 2019 were relevant due to the lack of historical satellite imagery prior to that date. This resulted in approximately 30 relevant events being identified.
2. Eplaz [51]: eplaz is an application that collects data on landslides in Slovenia. We requested and received data from 4 municipalities: Celje, Maribor, Vojnik and Žalec. After filtering to only include datapoints occurring in 2019 or later, approximately 150 events were collected. However, after the initial checks, it became evident that most landslides are small in scale and occur beneath tree cover, making them unobservable in satellite imagery.
3. Zbirka Podatkov o Graditvi objektov: Data collection of building construction in Slovenia. We selected 2 examples of new construction.

Satellite imagery was obtained by setting up a trial agreement with PlanetLabs. We obtained a few samples of SkySat and PlanetScope imagery to compare. SkySat has limited archive availability, meaning we were not able to receive a before-and-after image for any of our historical events. PlanetScope does however have extensive archives available. We received 20sqkm of download quota from the PlanetScope images. We were then able to download before-and-after images covering an area of approximately 10 km².

To ensure the quality of the validation process we first manually reviewed some of the historical events through GoogleEarth history maps to discard false datapoints, or events that cannot be observed. At the same time, we estimated the date of occurrence of the events. In total, 20 pairs of PlanetScope images were downloaded: 8 corresponding to illegal build-up, 2 to new construction, and 10 to landslides. The samples were analysed through visual inspection and evaluated using our LDM-CD model.

Some results can be observed in Figure 6. The left and middle images are satellite images taken at two points in time while the right image is the prediction with our LDM-CD model. Human activity of larger scale such as new construction (first row in Figure 6) is detected by the model, while smaller changes such as bales of hay (second row) cannot be visually observed. A similar situation applies to landslides (shown in the third and fourth row). However, the model detects the change with much lower confidence. Landslides are more difficult to detect than buildings, as they represent natural changes that can easily blend into the surrounding environment.

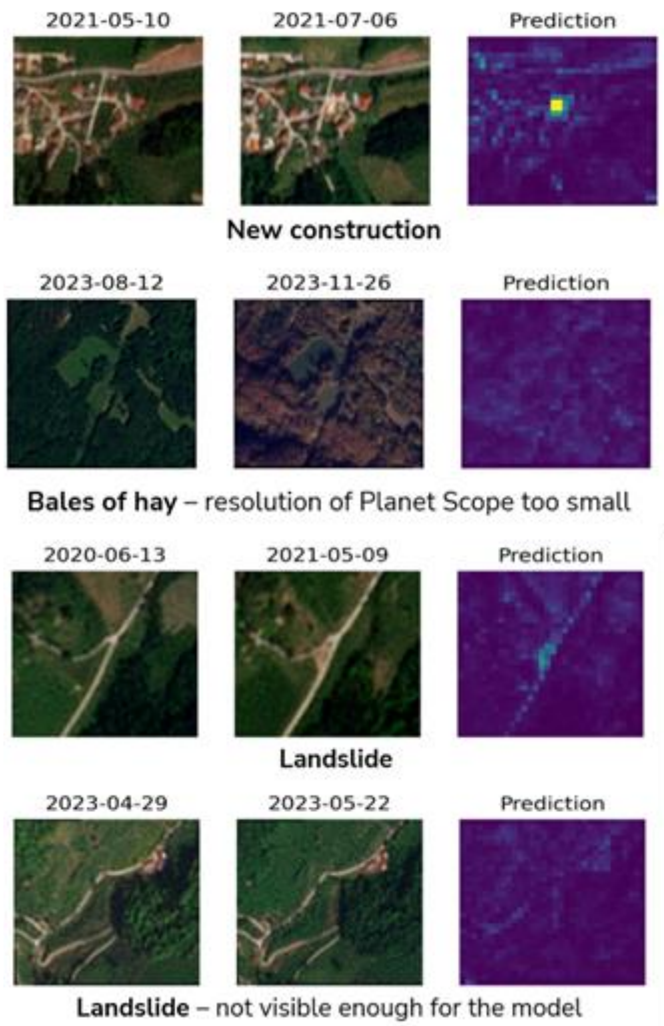

Figure 6: Change Detection module with PlanetScope imagery. The left and middle image show the before and after satellite image, the right shows the detection with the change detection model. The upper two rows show human activity (new construction and bales of hay) the lower two show landslides.

An additional challenge encountered during validation was the presence of shadows. The SkySat imagery contained significant shadows, which were incorrectly detected as changes. Due to the very limited number of SkySat samples, we were therefore unable to validate the method effectively. .

Another issue we observed was setting a threshold for the detected change. The model gives a confidence score between 0 and 1, which then requires tuning a threshold to obtain a binary classification result. As evident in Figure 6, some changes are detected with much lower confidence than others. We were therefore unable to calculate the threshold at this stage. It is expected that the threshold will be tuned during the operational use of the tool, allowing users to review the predictions and assess the relevance of the detected changes based on their specific needs.

Pilot 2

In Pilot 2 we will continue the validation of the tool, with a particular focus on the ELS use case. We will continue with the validation of both medium (3m) and high (50cm) resolution imagery. Satellite imagery is collected as follows:

1. Medium resolution imagery: We obtained access to PlanetScope imagery through SentinelHub. Our target area is 100km of powerline with 200m wide corridor, resulting in an area of 20 km². We tried to include various landscapes (buildings, forests, fields) and powerlines with known historical issues as observed from ELES illegal buildup dataset. We selected 2 powerlines: 2x400 kV Beričevo-Okroglo and 220 kV Kleče-Divača resulting in total of 96km of powerlines. Additionally, there are 5 historical illegal buildup events along the two lines, which will serve for further validation. We gained access to all historical and current PlanetScope imagery over the defined area for the next year.
2. High resolution imagery: Most high-resolution satellites operate on a tasking basis. However, since change detection requires before-and-after imagery, we require historical imagery. As previously noted, the archive of available SkySat imagery is very limited. SkyFi[60] is a platform combining geospatial data from multiple satellite providers. An alternative to SkySat satellite was Siwei satellite, which also has 50cm spatial resolution but with a more extensive archive. We mapped the unused ELES illegal buildup datapoints to the Siwei historical archive and identified 4 potential events. For each event, we downloaded between 3 to 4 optical satellite images captured in separate years.

We will validate the collected imagery using our method. Additionally, the imagery is included in the deployed solution to be used as part of Pilot 2.

2.3.2 Vegetation monitoring

The second component of our tool is vegetation height prediction, which supports vegetation monitoring. Traditional vegetation monitoring relies on LiDAR imagery, which is costly and infrequently available. Satellite imagery offers an alternative, which is more cost effective and can provide continuous monitoring over a large area. Our solution uses a supervised machine learning model trained on extensive labelled data to predict the height of vegetation from multispectral satellite imagery. In D7.2[46] and D7.3[47], we provided a detailed description of the techniques used and in D7.4[48], we described the tool validation as part of Pilot 1. In this deliverable, we provide a summary of the methods and data used in our solution and the validation of the component.

2.3.2.1 Vegetation Height Data

Our solution is based on generating predictions from multispectral satellite imagery. Since the method used in the solution is a supervised model, it was necessary to use the same type of imagery both during the training process and in the inference stage of the deployed solution. When selecting appropriate satellite imagery providers, we therefore needed to consider both the ability to train the model and the ability of use for end users. The main criteria were therefore cost, availability of historical archives (due to the extensive training data requirement) and spatial resolution. We identified ESA Sentinel-2 and Planet PlanetScope as having appropriate historical archives available for our purposes. Sentinel-2 offers multispectral imagery with spatial resolution ranging from 10 to 60 meters, along with a revisit time of 2-3 days. It is part of the European Space Agency's Copernicus Program and is freely available. PlanetScope is operated by Planet and provides imagery with a spatial resolution of 3-4 meters and a daily revisit time. In the project, we used on Sentinel-2 imagery due to its accessibility.

For vegetation height labels, we used the Vegetation Height Model (VHM) developed by the National Forest Inventory of Switzerland[16]. It includes a vegetation height data with 1x1 meter resolution across Switzerland. Our target areas are in Slovenia, which has similar vegetation characteristics to those in Switzerland. We therefore identified the VHM data to be suitable for model training and expect the trained model to generalize well-targeted regions.

We matched VHM data with the corresponding Sentinel-2 imagery and pre-processed the data to filter out clouds and water bodies. We split the data into training, validation and test set. Additionally, we used data transformations and added features such as Normalized Difference Vegetation Index (NDVI).

2.3.2.2 AI Method

Vegetation height prediction involves making pixel-wise predictions, making it closely related to the computer vision task of semantic segmentation. The main difference is that semantic segmentation makes categorical prediction (i.e. predicts a class) while vegetation height predicts a continuous value. For this reason, we used the existing segmentation models as our basis.

As noted, models like change detection models have a backbone and a head component. The backbone is responsible for feature extraction from input images. We experimented with the architectures of several segmentation models including Unet [2], DeepLabv3 [17], Swin Transformer [18], ConvNeXt [19], Riad [3], Feature Pyramic Netword [4] and EfficientNet B8 [1]. EfficientNet B8 proved to be the best-performing model for our task. Head component uses the extracted features to make predictions – this is where our models differ from segmentation models.

We used Mean Absolute Error (MAE) as the training loss and evaluation metric to measure model performance.

During model training, we normalized input data to mean 0 and standard deviation 1, and scaled vegetation height to a range of [-1,1]. Models were trained with an AdamW optimizer and batch size of 32. Additionally, augmentation techniques such as flipping and rotating were applied during training. Evaluation was only done on valid pixels which do not overlap with snow or water.

The Unet model with an EfficientNet B8 backbone reached an MAE of 1.6 meters. The RIAD model showed similar performance, but we chose EfficientNet as our end model due to its extensive use in computer vision tasks.

2.3.2.3 Tool Validation

The relevant CI partners identified for the vegetation management solution are ELES and SZ. Validation of the solution requires making predictions at the target site and comparing it with the ground truth.

Traditionally, LiDAR imagery is used for vegetation management and measuring the height of trees. Aerial LiDAR imagery is highly accurate and therefore can be used as ground truth to evaluate our alternative solution with satellite imagery.

We received two LiDAR samples to validate our solution: 1) a sample captured during Pilot 1 flights over SZ area in 2024, and 2) a sample provided by ELES from 2017.

The LiDAR Point Cloud sample was first converted to Vegetation Height Model (VHM) using Whitebox Workflows[61]. From the point cloud, we first computed the Digital Terrain Model (DTM) – a model of the bare surface – by modelling points corresponding to the last return and filtering out points that did not correspond to the ground. Secondly, we computed the Digital Surface Model (DSM) – a model of ground, canopy tops and building roofs – by modelling first return points. VHM is then computed as VHM = DSM – DTM. The resulting model has 1m spatial resolution, which is sampled to 10m to match the prediction based on Sentinel imagery. An example of LiDAR, DSM, DTM and VHM images can be observed in Figure 7.

We downloaded Sentinel satellite images within 6 months of LiDAR capture date, used the pretrained model to make predictions and compared them to the ground truth. We used MAE as the metric to compare both vegetation height models.

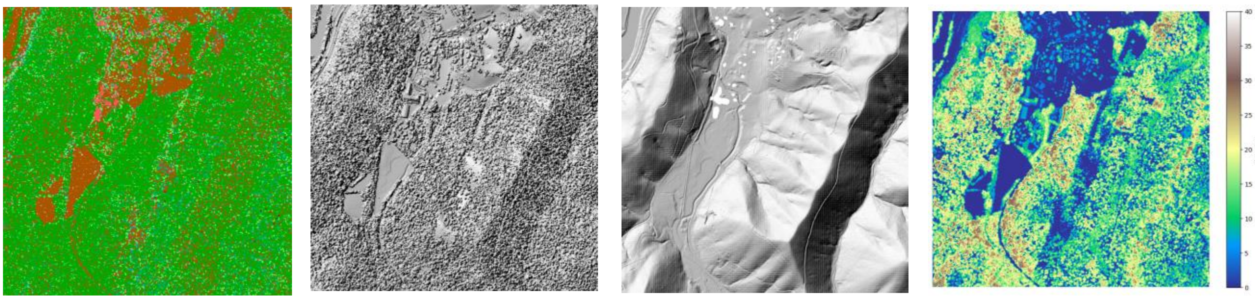

Figure 7: Converting LiDAR sample to Vegetation Height Model. The images from left to right are LiDAR, Digital Surface Model, Digital Terrain Model, Vegetation Height Model.

Sample 1 was captured during the Pilot 1 trials in June 2024. It captures a 700m long 120m wide area. 4 suitable Sentinel images were available in the 6m timeframe around the capture date. The average MAE of the prediction was 4.21m. Note that the high error is most likely due to inaccuracies in the LiDAR sample that was further from drone's flight line, since that is also where the error was the highest. Results can be seen in Figure 8.

Sample 2 covers a much larger area (8km long and 170m wide corridor). The image was captured in 2017, and 5 corresponding Sentinel images were available for download. The MAE between the prediction and the ground truth was 2.8m. Results over multiple sentinel images can be seen in Figure 9.

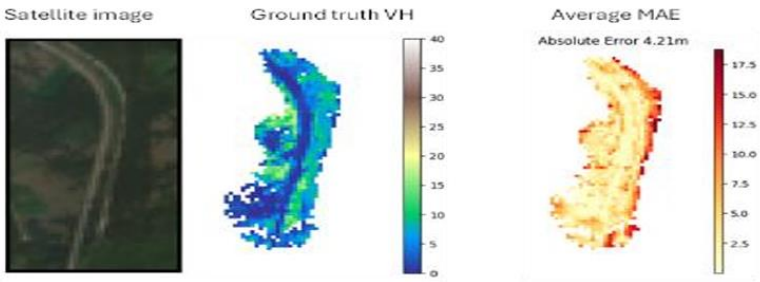

Figure 8: Sample 1 results. Left: Sentinel imagery, Middle: Ground truth vegetation height; Right: MAE of average prediction.

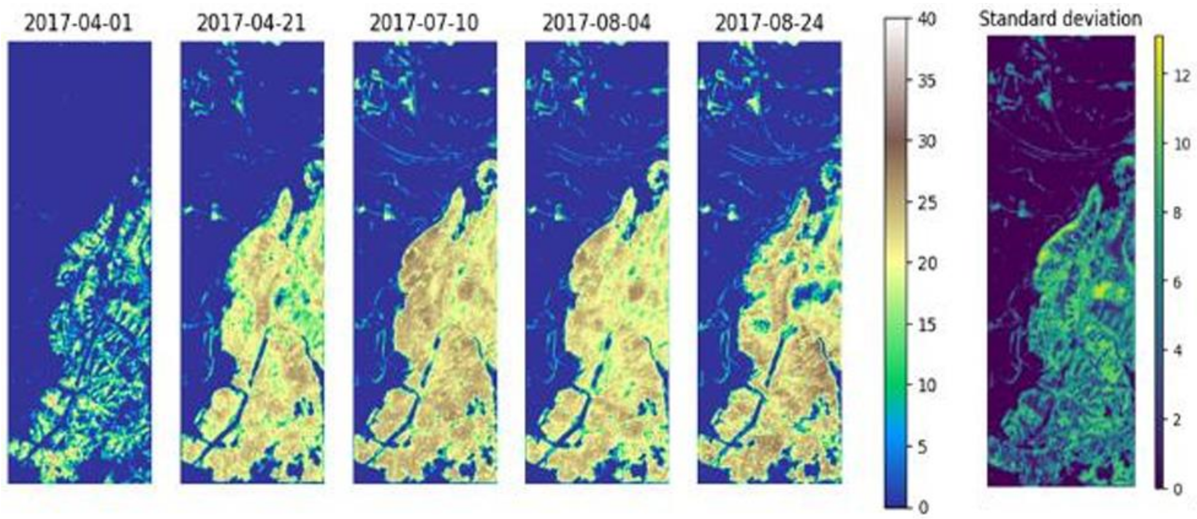

Figure 9: Predictions, using the vegetation height model, for all 5 available Sentinel images

The validation phase allowed us to gain more insight into how the tool could be used. Some important discoveries were:

- The absolute error is small for low trees but high for taller canopy. The model is therefore good at determining the presence of trees but lacks accuracy when estimating taller vegetation. Since the number of trees taller than 25m is very small, the high error at higher vegetation is less relevant.
- Due to the low revisit time (cca. 5 days) of Sentinel satellites and bad weather conditions in the target areas, there are approximately 1-2 appropriate images available from the summer months.
- The model was trained on the data from the summer months, so it can only be used during those months.
- Using the average prediction over multiple satellite images produces better results, since the atmospheric conditions could cause some predictions to be less accurate.

These insights allow us to estimate that our solution could be used to calculate vegetation height biannually.

2.4 Deployment

Our solution is available as-a-service and is accessible via a REST API, which allows for easy integration with the interface developed in T7.3 and with any other pilot legacy system. Only the inference part of the solution is deployed, since the models are static and do not need retraining. The solution is containerized using Docker[62] and deployed in the Open Stack[63] cloud. REST API was implemented using FastAPI[64].

The API enables change detection and vegetation height prediction at two different endpoints, as shown in Figure 10.

The solution was deployed with Sentinel-2 satellite imagery and with limited samples of PlanetScope and SkySat imagery. For full integration of PlanetScope imagery, the user would need to acquire a subscription of the satellite imagery from Planet.

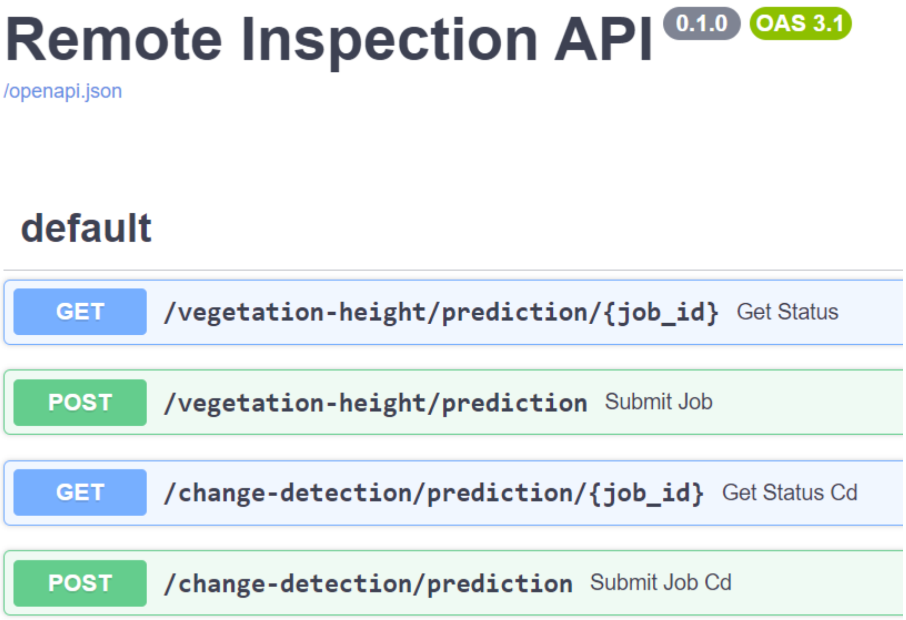

Figure 10: API endpoints of the satellite inspection tool.

2.5 Applicability of the satellite imagery-based tools for the pilots

We invested significant resources in research and bilateral meetings with CIs and investigated their needs and requirements. We chose this approach in order to guarantee the applicability of the toolset from a usability and financial standpoint – we as tool developers must ensure that the developed functionalities generate impact and add value to the CIs.

These entailed meetings were conducted not only with personnel directly involved in the project, but with more operative personnel as well, who are directly connected to fieldwork. Here, we discuss the applicability of our methods for the specific problems of the pilots.

2.5.1 HDE

The execution of additional discussions with HDE proved that the viability of using satellite imagery is very limited. Namely, HDE is interested in detecting clogged grates, presence of excessive sediments in riverbeds near the weirs, problems with their dams, etc. We explored this idea and pursued satellite imagery of the highest resolution, obtained from Airbus Pléiades Neo constellation. However, after the evaluation by HDE, these satellite images, despite their high resolution of 30cm, are still unsatisfactory for such purposes. Furthermore, obtaining such images for commercial use would also result in a high cost of the method; although they were provided to us free of charge their cost is estimated to be in the range of 50,000 EUR. Deploying a solution with such a high cost would result in a recurring financial burden for HDE that they are unwilling to incur, even if the imagery was of satisfactory resolution.

2.5.2 SZ

The vegetation height model developed on the publicly available data [16] works well and has been validated using the dataset itself as ground truth. However, we further investigated the requirements of the SZ before we conducted a full-scale validation, which also requires additional investment from their side. The legal framework of the Republic of Slovenia defines 100-meter-wide area along the tracks is designated as a no-build zone, where neither buildings nor trees are permitted. (Zakon o varnosti v železniškem prometu - ZVZelP-1 - Article 26). All the special cases (e.g. when the tracks are in a narrow valley) are individually evaluated and risk analysis is carried out. From this perspective, SZ finds it borderline useful to implement a fully-fledged model now. However, the model and the work done is accepted and ready to be used if circumstances change.

Additionally, SZ was interested in the change detection module – specifically detection of imminent landslides. Together, we reached the conclusion that detection of landslides using just satellite imagery is not useful. They are, quite simply, detected faster using traditional tools, observers or even trains because satellites have relatively low spatial resolution.

2.5.3 ELES

Similar to SZ, we evaluated the usefulness of the developed vegetation height model with ELS. They have similar legal protection (Energetski zakon (EZ-1); Article 468) which defines a 40-meter-wide band on each side of the power lines, where no buildings or vegetation (including trees) are allowed. Given this area is free of trees, we considered a variant where there is fast-growing vegetation that could potentially present a concern. However, such cases are practically non-existent, rendering this concern a non-issue. Furthermore, we should account for the changing height of the power lines, which could present a security risk. Given the reasoning and low interest in this pilot, we decided the vegetation height is not a concern for ELS.

On the other hand, an issue was identified with illegal build-ups. In practical terms, landowners may decide to build structures in legally protected areas or just fence off the land. A particular concern is posed by hunting lodges, which are typically generally tall and feature metal roofs. In this case, the ELS is required to intervene. Detecting such build-ups is highly valuable for the ELS, which led us to jointly decide to pursue this direction in our modelling, with a focus on change detection.

2.5.4 ACO

Additional talks with ACO revealed that the satellite imagery can detect large spills of water - typically if they are large enough to create a visible stain on the surface. However, such large spills are detected much sooner using the technology ACO already has, which takes advantage of the direct control of water pressure in the water supply pipes. Furthermore, it requires the event to be in remote areas as pipe spills and bursts in urban areas are immediately detected. As a result, we jointly decided to pursue a more cost-effective and viable technology: UAVs.

3 UAV inspection tool

This section presents the final version of the UAV-based inspection tool developed within the SUNRISE project, a journey that begins with the conceptualization of the tool in D7.1 [45]. As this is the final iteration of the Infrastructure Inspection Tool and Training Guide, this section aims to provide a self-contained overview of all the work carried out on UAV inspection, integrating key developments from previous deliverables while maintaining conciseness. References to previous versions, D7.2 [46] and D7.3 [47], will be included when necessary to highlight technical progress or to avoid delving too deeply into previously documented materials, but the goal is to minimize redundancy while ensuring clarity and completeness.

The UAV inspection tool has been designed to enable remote, AI-assisted assessment of critical infrastructure, leveraging major advances in **computer vision and multimodal large language models** (M-LLMs) (also referred to as vision-language models (VLMs)) to address real-world challenges in an innovative and effective way. This is reflected in its three main modules: object detection, semantic segmentation, and Visual Question Answering (VQA), which provide automated analysis of a broad range of diverse infrastructure conditions. Additionally, secondary modules such as 3D virtualization and anonymization enhance the tool's capabilities while ensuring compliance with privacy regulations and user data protection.

This final version consolidates previous developments by integrating lessons learned from pilot trials, documented in D7.4 [48], and refining the system's architecture, processing models, and deployment strategies. Emphasis has been placed on bridging the gap between end users and the tool functionalities by implementing a user-friendly GUI that enhances accessibility and usability. The document provides a comprehensive overview of the high-level architecture, core tool modules, validation processes, and final integration of the UAV platform. Special attention is given to the technical improvements driven by real-world testing, ensuring that the tool meets the operational needs of CI operators and is fully prepared for real-world deployment.

The structure of this section is as follows:

- General context and architecture: A recap of the tool's purpose and its integration within the broader infrastructure inspection framework.
- Tool modules and AI functionalities: Final descriptions of the object detection, segmentation, VQA, anonymization, and 3D virtualization components.
- Validation and performance assessment: Results and conclusions drawn from laboratory and field testing.
- Final deployment considerations: Practical aspects of integrating the UAV tool into real-world operational workflows.

By providing a comprehensive yet streamlined summary, this section ensures that readers can understand the full scope and impact of the UAV inspection tool without requiring excessive cross-referencing to previous deliverables.

3.1 General context

The UAV Inspection Tool is a key component of the SUNRISE remote inspection framework, designed to modernize the **monitoring and maintenance of critical infrastructures (CIs) through AI-driven automation and high-resolution aerial imaging**. By integrating computer vision techniques, multimodal large language models (M-LLMs) and real-time data processing when needed, this tool enhances the ability of infrastructure operators to conduct non-invasive, frequent and risk-based inspections, minimizing reliance on traditional manual assessments.

Developed under WP7, the UAV Inspection Tool plays a crucial role in complementing satellite-based monitoring, offering detailed, localized inspections where satellite imagery lacks the necessary resolution. This combination of wide-area surveillance and precise UAV-assisted analysis provides a multi-layered inspection strategy, optimizing resource allocation to enable timely maintenance interventions. The tool's modular and adaptive architecture allows UAVs to be configured with different sensor types, adapting to the specific needs of energy, transportation and water infrastructure sectors.

The UAV Inspection Tool addresses major challenges in infrastructure monitoring, such as aging assets, environmental risks and regulatory compliance requirements. Traditional inspection methods are often time-consuming, costly and constrained by personnel availability, limiting their frequency and effectiveness. By automating key inspection processes, the UAV tool enables the early detection of structural defects, vegetation encroachment and environmental hazards, significantly reducing operational risks and optimizing maintenance workflows.

Its applications span multiple critical infrastructure domains, including power grids and railway catenary networks, where it helps to monitor insulator conditions, detect corrosion and assess vegetation threats. It is also used for environmental risk detection, evaluating the impact of floods, landslides and wildfires in real-time, as well as for structural integrity assessments, identifying cracks, leaks and material degradation in bridges, tunnels and pipelines.

By integrating UAV-based imaging with AI-powered analytics, **the tool strengthens infrastructure resilience, operational efficiency and cost-effectiveness in maintenance workflows**. The SUNRISE project's holistic approach ensures that this technology is not only a solution for crisis scenarios but also an essential component of everyday infrastructure management, improving safety, longevity and sustainability.

3.2 Architecture: High level design & use cases definition

The UAV Inspection Tool has been developed following three key design principles: **self-containment**, **modularity** and **scalability**. These principles ensure that the tool can operate independently, integrate seamlessly with other inspection modules within the SUNRISE remote inspection framework, and scale to meet increasing computational and data processing demands.

The **self-contained** nature of the UAV Inspection Tool ensures that it functions independently, minimizing dependencies on external components and isolating its codebase to prevent conflicts between library versions. This ensures stable and reliable performance across different deployment environments and allows for continuous operation without being affected by changes in other parts of the system. The **modular design** facilitates the addition, removal or modification of components without disrupting the system's overall functionality, allowing for future expansions or improvements in specific inspection capabilities. Given the rapid evolution of AI models, where new releases frequently surpass their predecessors in performance and efficiency, this modularity is crucial. It ensures that upgrading or replacing AI models does not require extensive reconfiguration of the entire system, allowing the tool to remain at the forefront of technological advancements without compromising stability.

To achieve this, the UAV Inspection Tool is structured around a **REST API-based architecture**, ensuring standardized communication between its components and seamless integration with other elements of the SUNRISE platform, including the WP7 general GUI. The system is designed for flexible deployment, supporting both cloud-based processing for large-scale batch analyses without strict time constraints and edge computing for real-time inspection scenarios where immediate processing is required.

Additionally, **Dockerization** has been implemented to containerize the software components, ensuring consistent performance across different computing environments. This encapsulated deployment approach simplifies system updates, **scalability** and portability, making it easier to deploy and manage the inspection tool across multiple infrastructure sites. Software updates are centrally managed via a GitHub repository, ensuring distributed version control and seamless synchronization across processing environments without service disruptions.

The core architecture consists of four main entities:

- **Processing and Analysis Module:** Hosted either in the cloud or on an edge device, this module runs the AI-based models responsible for object detection, semantic segmentation and VQA-based infrastructure assessment.
- **Global Infrastructure Monitoring GUI (WP7):** A centralized interface where alerts from various inspection tools are displayed, providing a global overview of the CI's status and allowing operators to track ongoing incidents and manage risk at a macro level.
- **UAV-Specific Inspection GUI**: A dedicated interface for UAV operations, where CI operators can launch inspection processes, both in real-time and using pre-recorded imagery. This GUI provides direct control over UAV-based monitoring, offering detailed visualization and task execution management at a local level.
- **UAV Platform:** The data acquisition unit, equipped with various sensors and cameras to capture high-resolution imagery for inspection tasks.

These four entities and their interactions are defined in Figure 11.

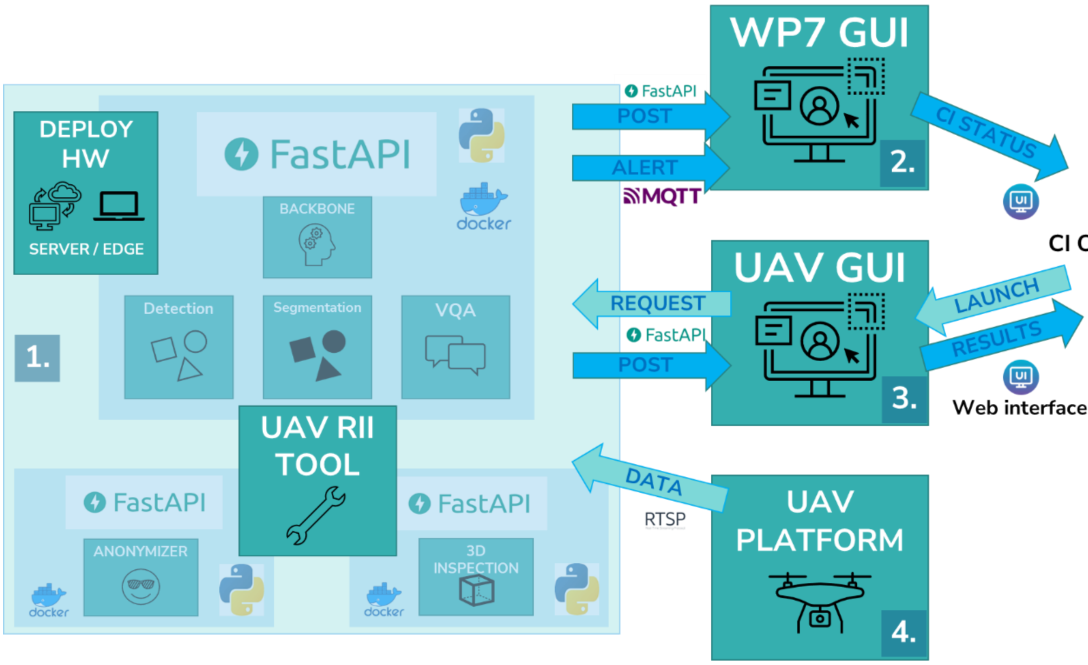

Figure 11: UAV RII tool architecture. 1- Processing and Analysis Modules; 2- Global Infrastructure Monitoring GUI (WP7); 3- UAV Specific Inspection GUI; 4- UAV Platform.

Communication between these components is managed via MQTT messaging and API calls, allowing efficient data exchange and inspection requests. The communication between the UAV and the data analysis hardware is carried out either via RTSP when streaming video in real-time or by downloading the data directly from the UAV's memory card after the flight, depending on the operational scenario and inspection requirements. The CI operator initiates the inspection process from the UAV-specific web interface, where they can also visualize the results once the analysis is complete. At the same time, they can track the overall status of the infrastructure through the global alert system web interface, allowing them to either examine detailed UAV inspection results or receive high-level alerts and reports in the WP7 general tool, depending on their operational needs.

The hardware configuration of the system is adaptable, allowing deployment across multiple processing environments. The UAV Inspection Tool can run on dedicated edge devices, central servers or hybrid infrastructures, depending on the operational scenario. In Pilot 1, for example, the system was deployed across multiple hardware instances simultaneously: a Jetson Orin device was used to support real-time detection processing, while the batch processing service remained operational on the central server. This distributed architecture enables optimized workload balancing, ensuring that real-time capabilities are maintained without compromising large-scale analytical tasks.

As noted, the system supports two primary operation modes: **live-streamed UAV inspections**, where analysis is performed in real time on edge devices, and **batch processing**, where pre-recorded UAV imagery is analyzed using cloud-based computing resources. This dual approach provides deployment flexibility, allowing CI operators to choose the most suitable method based on their specific operational requirements. Figure 12 and Figure 13 illustrate a flowchart with an example use case for each mode, following the same component nomenclature as Figure 11 to ensure a clear and cohesive understanding of the overall workflow.

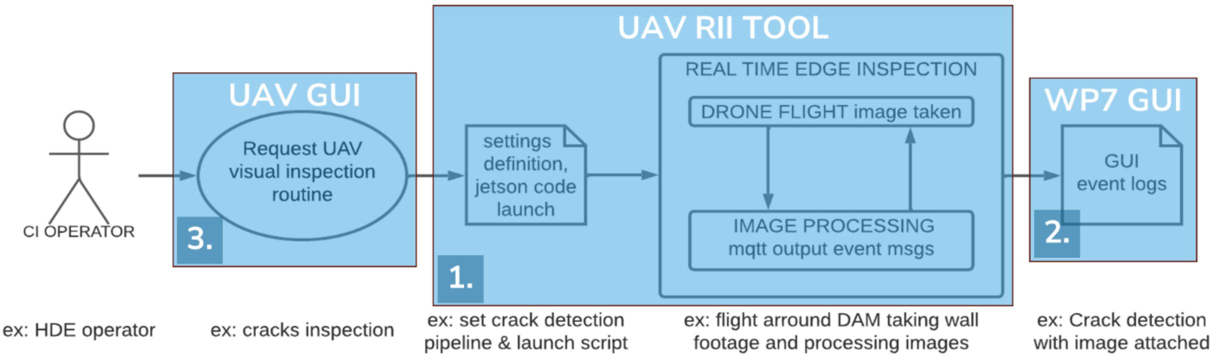

Figure 12: UAV remote inspection tool live stream processing use case flowchart. 1- Processing and Analysis Modules; 2- Global Infrastructure Monitoring GUI (WP7); 3- UAV Specific Inspection GUI.

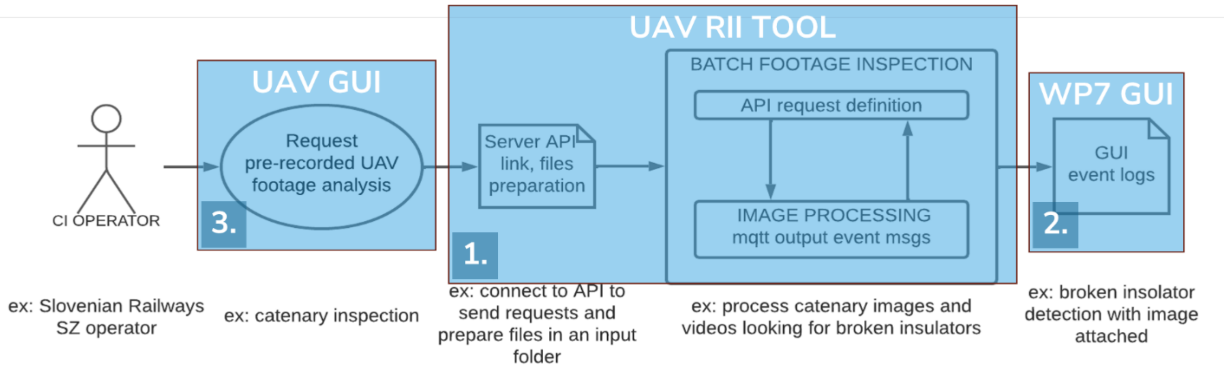

Figure 13: UAV remote inspection tool batch processing use case flowchart. 1- Processing and Analysis Modules; 2- Global Infrastructure Monitoring GUI (WP7); 3- UAV Specific Inspection GUI.

In summary, the UAV Inspection Tool has been designed to seamlessly integrate into the broader SUNRISE ecosystem, enabling real-time and large-scale inspections through a robust, modular and scalable architecture. By leveraging cloud-native principles, API-driven communication and flexible deployment options, the tool ensures enhanced operational efficiency, adaptability, and ease of integration for CI operators.

3.3 Tool modules description

The UAV inspection tool retains the modular architecture established in previous iterations, structured around three main modules: **Detection**, **Segmentation** and **Visual Question Answering (VQA)**; and two supplementary modules: **3D Virtualization** and **Anonymization**.

The main modules focus on the core inspection tasks. The Detection module identifies critical infrastructure components within UAV-captured images or detects real-time anomalies such as fires. The Segmentation module extracts key visual elements by accurately delineating the inspected objects/structures and removing background noise, enhancing the clarity of inspected features and enabling semantic segmentation of visually relevant concepts. The VQA module enables interactive AI-based queries, allowing for classification, contextual understanding and automated interpretation of the inspection data.

The supplementary modules extend the tool's capabilities beyond core inspection. The 3D Virtualization module reconstructs high-fidelity structural models from UAV footage, facilitating spatial analysis and comparative assessments over time. The Anonymization module ensures compliance with privacy regulations by automatically detecting and obfuscating sensitive information within captured images, such as human faces or identifiable objects.

This modular approach, reflected in Figure 11 in the previous section, ensures flexibility and adaptability, allowing different configurations of the core inspection modules based on the inspection scenario. This versatility and reconfiguration capability are illustrated in Figure 14, which showcases all possible pipeline configurations that can be created to tailor the solutions by using the modules in single mode, in a cascade (one's output as another's input), or all at once independently.

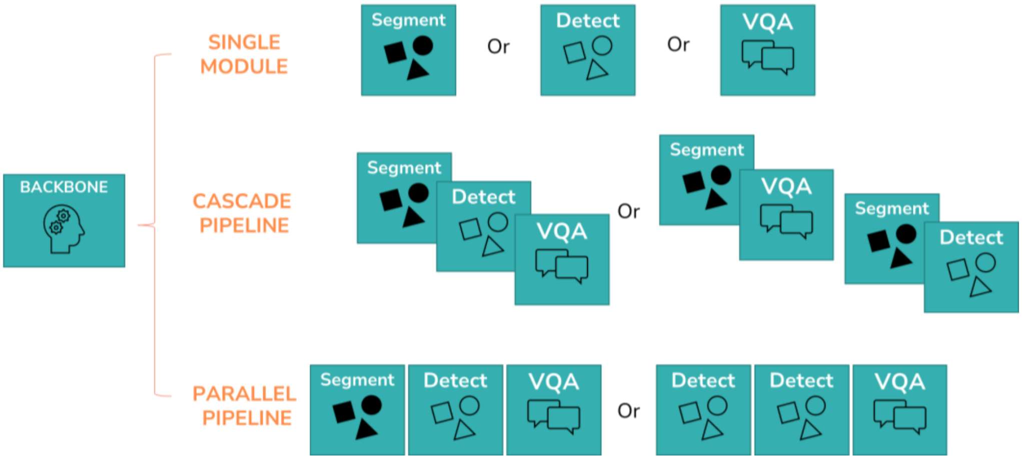

Figure 14: Dynamic pipeline generation.

While pipelines can be dynamically reconfigured, each of the modules within them is also highly customizable. Each module supports multiple AI models tailored to different scenarios and use cases, offering both predefined options, featuring models trained within the project, and open-vocabulary configurations, where users can create entirely new inspection pipelines beyond the initially defined ones. Figure 15 provides a detailed breakdown of the main modules and the AI models integrated within each. Further details of each module are provided in the subsequent subsections.

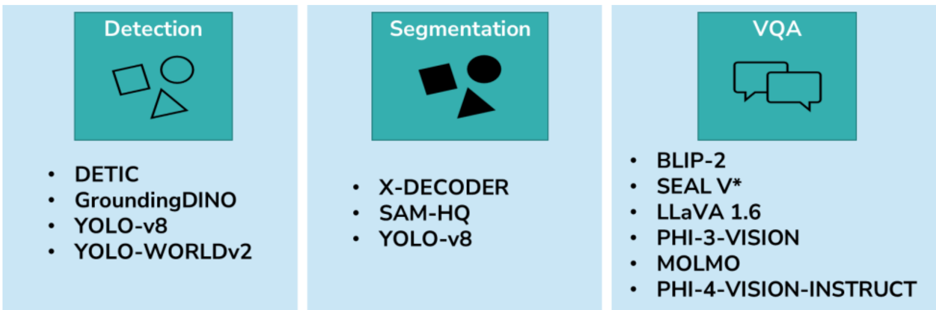

Figure 15: Main API tool modules models composition.

In this final phase, efforts have been directed towards refining the integration between modules, ensuring seamless interoperability across the inspection pipeline. **Compared to the previous** iteration (D7.3[47]), the **current version incorporates the validated outcomes from the first real pilot trials (Pilot 1)**, as documented in D7.4[48]. Significant advancements include the introduction **of more advanced models within the VQA module**, enhancing its ability to process and interpret inspection data with greater accuracy and contextual awareness. Additionally, considerable improvements have been made to the **user interface design**, simplifying the configuration of dynamic pipelines through the UAV tool's GUI and making it more intuitive for users to adapt and customize the inspection workflow according to their specific operational needs.

3.3.1 Object Detection and Semantic Segmentation

The detection and segmentation modules are fundamental components of the UAV inspection tool, enabling the identification, classification and delineation of infrastructure elements and potential anomalies in UAV-captured images. This module plays a **dual role within the inspection workflow**: it can function as a **standalone event detector**, identifying critical issues such as floods, fires or structural damage (cracks, corrosion), while also serving as a **pre-processing step** for higher-level functionalities, such as VQA-based inspection and anomaly assessment.

To provide both flexibility and high accuracy, the detection and segmentation module follows a dual approach, integrating two complementary methodologies:

- **General-purpose, open-vocabulary models**

  These models offer broad applicability across different inspection scenarios without requiring prior retraining. They leverage text-based queries to detect objects dynamically, enabling adaptive inspection pipelines that can be adjusted to new use cases. This approach ensures scalability, as new detection tasks can be introduced without the need for additional dataset collection and fine-tuning.

  The models included in this category are: Detic [27], X-Decoder [28], GroundingDINO [23], SAM-HQ [24] and YOLO-Worldv2 [26].

- **Dedicated, high-performance models**

  These models are optimized for specific infrastructure inspection tasks where high precision is required. Examples include corrosion detection, crack identification, and fire detection, which demand tailored models trained in domain-specific datasets. These models ensure robust detection performance, minimizing false positives and false negatives in mission-critical inspections.

  The only model included in this category is YOLOv8 [25], which has been retrained for three key previously mentioned tasks: fire detection, crack detection and segmentation, and rust detection and segmentation.

By combining these two approaches, the UAV inspection tool achieves a balance between adaptability and reliability, making it suitable for a wide range of real-world infrastructure monitoring applications.

A more detailed analysis of each general-purpose **open-vocabulary model** and its specific characteristics and metrics, along with extensive use examples in our data domain, was already provided in D7.2 [46] and D7.3 [47]. Open-vocabulary models refer to deep learning models capable of detecting and segmenting objects without being explicitly trained on a predefined set of classes. Instead of relying on fixed labels, these models leverage text-based queries to dynamically identify objects in images, allowing greater flexibility and scalability for diverse inspection scenarios. For example, models like GroundingDINO [23] and YOLO-Worldv2 [26] can detect objects such as "corroded metal" or "damaged insulator" based on descriptive prompts rather than rigid category labels.

It is important to highlight that no new models have been introduced in this category or any other category within this section since the publication of D7.3. Instead, the focus has been on validating their real-world applicability through Pilot 1. The results confirmed their effectiveness in infrastructure inspection, particularly in detecting structural anomalies in UAV-captured imagery, reinforcing the robustness of the previously selected approach.

The following content will focus on presenting the **results and performance metrics of the ad-hoc trained models** developed within the framework of the project. While these models have already been documented in previous deliverables, their inclusion here is essential to provide a comprehensive reflection of the tool's development process. For this reason, some figures from previous reports will be reused.

To evaluate the three ad-hoc object detection and segmentation models, a set of well-established metrics has been used to assess their accuracy and robustness. These metrics allow a balanced measurement of model performance by considering **true positives (TP)**, **false positives (FP)**, and **false negatives (FN)**.

The primary metrics used and formalized mathematically in Equation 1 are:

- **Precision**: Measures how many of the detected objects are correct.
- **Recall**: Evaluates how many of the actual objects were correctly detected.
- **F1 Score**: The harmonic mean of precision and recall, balancing both aspects to provide an overall measure of accuracy.
- **Intersection over Union (IoU)**: Measures the overlap between the predicted detection and the ground truth.
- **mAP (mean Average Precision @0.5 IoU)**: The most relevant metric for detection tasks, representing the average precision for all detected classes with an IoU threshold of 0.5, meaning a detection is considered correct if the predicted bounding box overlaps at least 50% with the ground truth.

Equation 1: Precision (A), Recall (B), F1 (C) and IoU (D)

$$Precision = \frac{True\ Positives\ (TP)}{True\ Positives\ (TP) + False\ Positives\ (FP)} \quad \text{(A)}$$

$$Recall = \frac{True\ Positives\ (TP)}{True\ Positives\ (TP) + False\ Negatives\ (FN)} \quad \text{(B)}$$

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad \text{(C)}$$

$$IoU = \frac{Area\ of\ Overlap}{Area\ of\ Union} \quad \text{(D)}$$

The evaluation of these metrics on training datasets confirms that the three ad-hoc trained models—crack detection, corrosion detection and fire detection—achieve robust performance before real-world validation. Their effectiveness in real infrastructure inspections is further assessed in the Pilot 1 validation section. Given the limited timeframe, the project has prioritized the development of a broad-spectrum inspection tool, rather than focusing on large-scale data collection and labeling for specific tasks. Consequently, the collected data has been used mainly to assess the overall system performance, rather than for extensive training refinement.

The **crack detection model**, based on YOLOv8X-seg, was trained using datasets [21] and [22] from Roboflow. These datasets contain images representing the expected conditions in real inspections, ensuring a realistic evaluation despite the focus on tool development over extensive dataset generation. Initial training runs using 100 epochs exhibited signs of overfitting, as seen in Figure 16A, where an increase in loss values and a slight decrease in mAP 0.5 were observed. To counteract this, training was limited to 50 epochs, as shown in Figure 16B, resulting in better generalization and improved model reliability. The final model achieved an mAP0.5 of 0.874 and an F1-score of 0.82, confirming its effectiveness in detecting structural cracks in concrete elements.

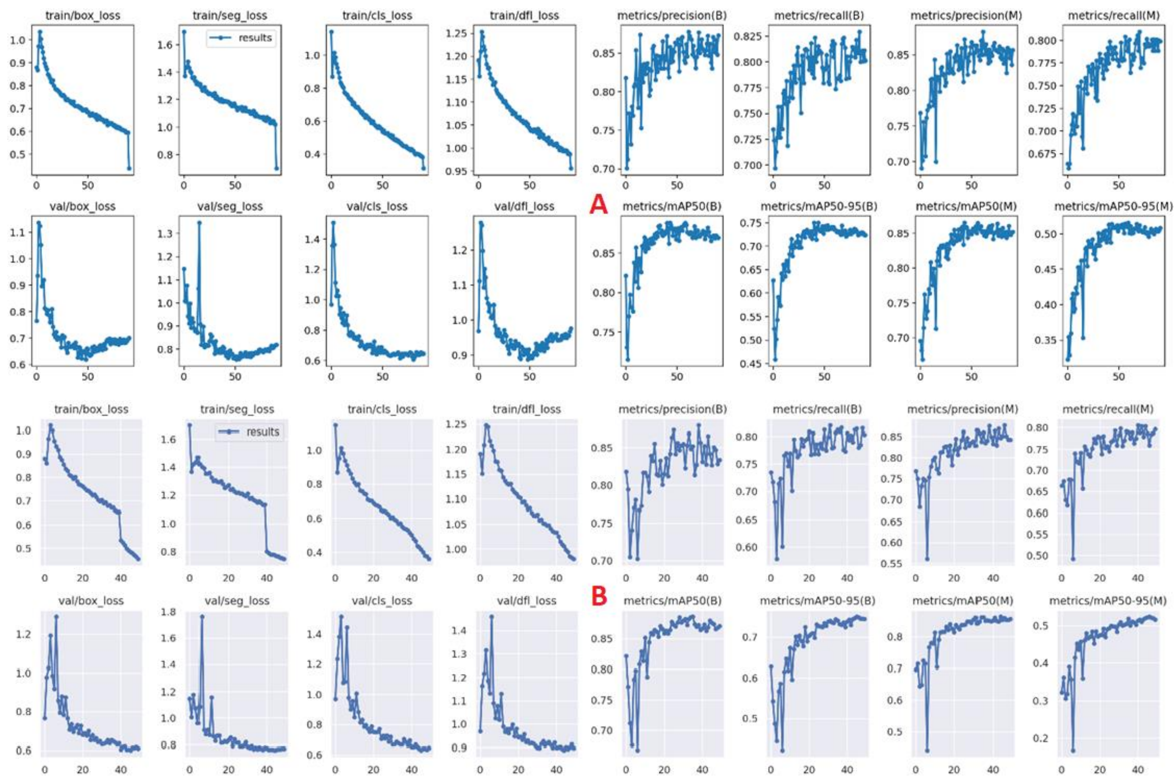

Figure 16: A (Upper) first iteration YOLOv8X-seg Crack detection model training metrics in combined datasets [20] and [21]. Signs of overfitting; B (Bottom) second iteration YOLOv8X-seg Crack detection model training metrics in combined datasets [20] and [21]. No sign of overfitting. Source: [47].

The **corrosion level detection model**, also utilizing YOLOv8X-seg, was trained to classify three levels of corrosion severity —fair corrosion, poor corrosion and severe corrosion— using dataset [22]. Unlike crack detection, this task proved more challenging due to the high variability in corrosion patterns and lower-quality datasets. As reflected in Figure 17, the model reached an mAP0.5 of 0.35 and an F1-score of 0.43. Despite this, the confusion matrix analysis showed that the model effectively identified 36% of severe corrosion cases, while maintaining a low false positive rate of 2%. These results indicate that while further improvements are needed, the model already provides valuable early-warning capabilities for corrosion detection, which could be refined with higher-quality labeled images collected from real-world pilots.

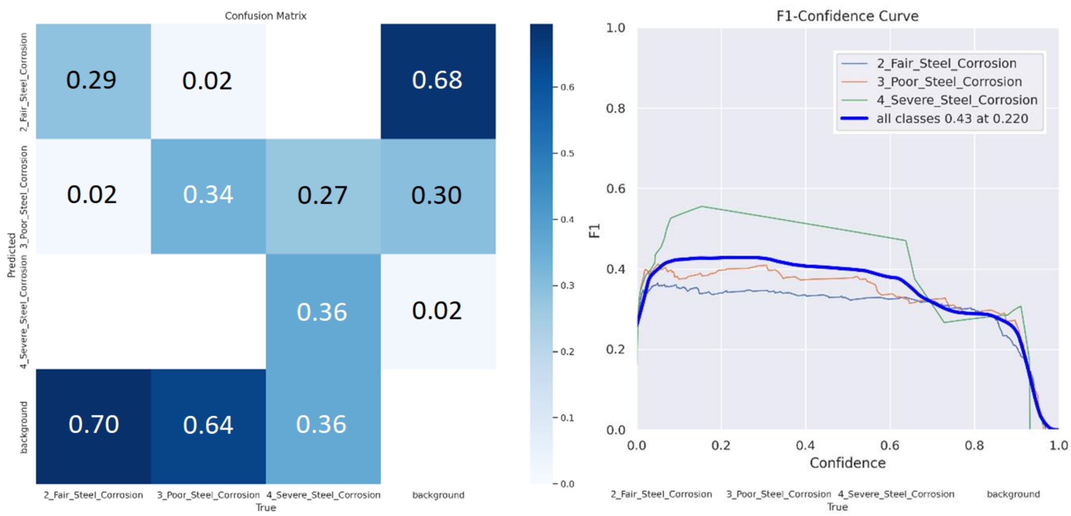

Figure 17: Confusion Matrix (left) and F1 curve (right) resulting from training YOLOv8X-seg in a dataset with three different levels of corrosion annotated, [22]. Source:[47].

The **fire detection model**, also built on YOLOv8 but for the detection task only, was trained using a dataset generated with Stable-Diffusion v2.1, employing synthetic data augmentation techniques to improve generalization. As seen in Figure 18, the model achieved an mAP0.5 of 0.64, demonstrating good reliability in detecting fire and smoke in UAV-captured images. This model was successfully validated under real-world conditions during the Slovenian Railways pilot, where it detected a controlled fire in real-time using UAV video feeds. The deployment on edge hardware (Jetson Orin) with a satellite-based local internet network (Starlink) confirmed its operational viability in remote locations, making it a valuable component of the inspection tool.

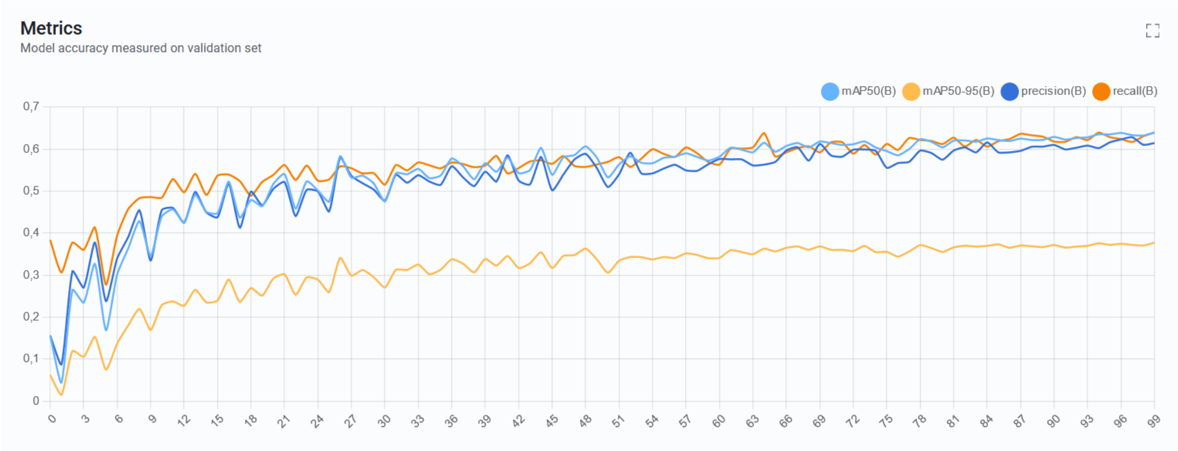

Figure 18: YOLOv8 fire and smoke detector mAP metrics. Source: [46].

These results, collected in Table 2, confirm that each of the ad-hoc trained models contributes significantly to infrastructure monitoring applications. The crack detection model stands out as the most mature and ready for deployment, achieving high precision and reliability. The corrosion detection model, while promising as an early warning system, requires further dataset refinement to improve its robustness across different conditions. Meanwhile, the fire detection model has already demonstrated its operational viability, successfully detecting fire and smoke in real-world UAV video feeds during Pilot 1.

Table 2: Summary ad-hoc trained YOLOv8 validation results.

Model	Task	mAP@0.5	F1-score	Notes
YOLOv8X-seg	Crack Detection	0.874	0.82	Best-performing model, robust results
YOLOv8X-seg	Corrosion Detection	0.35	0.43	Needs dataset refinement
YOLOv8	Fire Detection	0.64	0.625	Successfully tested in Pilot 1

At this stage of the project, the results reinforce the effectiveness of combining large foundational open-vocabulary models with task-specific models, confirming the adaptability of this hybrid approach for UAV-based infrastructure inspection. A key insight from these evaluations is that if strict real-time processing is not required, and a latency of approximately 10 seconds is acceptable, both types of models — detection, segmentation, and even VQA — can be integrated into a unified inspection pipeline. This enables a more robust and reliable detection system, where multiple models operate in parallel, cross-validating outputs to enhance accuracy.

While these results highlight the potential of our approach, operational validation is essential to confirm its effectiveness in real-world conditions. Section 3.4.1 details the findings from Pilot 1, where these models were tested in practical UAV inspection scenarios, providing key insights into their deployment and performance in field conditions.

3.3.2 VQA

The Visual Question Answering (VQA) module is a core component of the UAV Inspection Tool, designed to automate the interpretation of UAV-captured imagery through intuitive natural-language interactions. By leveraging state-of-the-art Visual Language Models (VLMs), this module offers unmatched adaptability, enabling operators to analyze and assess a wide range of inspection scenarios.

Unlike traditional AI classifiers, which require predefined categories and dedicated training for each specific task, VLMs allow flexible, on-the-fly evaluation of infrastructure conditions through open-ended queries. This capability enables operators to examine the status of critical components, assess structural integrity and analyze any visually relevant concept without the need for task-specific models. By **replacing or complementing classical AI classifiers**, the VQA module enhances scalability and responsiveness, making it a powerful tool for near real-time infrastructure monitoring across diverse operational environments.

The initial implementation of the module was built around **BLIP-2 [29]**, which was selected for being the state-of-the-art model at the time. It demonstrates reliable performance in visual inspection tasks, including anomaly detection, structural integrity assessments, and component evaluations such as power line insulators or clogged drainage grates, as reflected in the results presented previously in D7.2 [46]. BLIP-2 served as a robust baseline, combining high visual comprehension with computational efficiency, thereby laying a solid foundation for subsequent advancements.

To enhance context retention and region-specific analysis, in D7.3[47] the module was expanded to include **SEAL VQA (V*)[30]**, which introduced the Visual Working Memory (VWM) concept. This mechanism enables the model to selectively focus on relevant image regions, significantly improving interpretative precision, particularly in complex visual scenarios. As demonstrated in the previous deliverable, field tests at HDE facilities (evaluating safety cables) and Slovenian Railways (inspecting ceramic insulators) validated its effectiveness, clearly demonstrating its advantages in real-world infrastructure inspections.

The last model integrated and reported in the previous deliverable was LLaVA[31], a powerful open-source model that significantly improved the module's capability to analyze temporal changes in infrastructure. Specifically, **LLaVA-1.6-Next**[32] excelled in before-and-after comparisons, outperforming all models previously introduced, making it an invaluable tool for monitoring structural conditions. However, its substantial computational demands (34B parameters) limited deployment on edge devices and in resource-constrained environments, prompting the need for more efficient alternatives.

To address these challenges, recent efforts have focused on testing and integrating newer, more compact yet powerful multimodal models tailored specifically to the requirements of infrastructure inspection. The latest additions to the module include Phi-3.5 Vision[33], the recently released high-performance Phi-4 Multimodal-instruct[34], the Molmo-7B model [35], and the Qwen2.5-VL family (3B and 7B variants) [36].

Phi-3.5 Vision, developed by Microsoft, offers robust multimodal interpretative capabilities with reduced computational demands, facilitating effective and efficient real-time analysis on constrained hardware. **Phi-4 Multimodal-instruct** is an evolution of Phi-3.5 Vision, also developed by Microsoft, advancing its capabilities further by integrating text, vision, and speech processing within a compact yet powerful 5.6B-parameter model. Its modality-specific routing mechanisms enable efficient multimodal reasoning, which is particularly valuable for tasks involving visible spectrum analysis combined with thermal imagery. This is critical for detecting subtle infrastructural anomalies such as leaks, corrosion, or overheating components.

Molmo7B provides an excellent balance between computational efficiency and performance, enabling detailed visual reasoning and accurate textual responses within constrained hardware environments. Its streamlined architecture makes it highly suitable for near real-time, edge-based infrastructure inspections, complementing other models by effectively handling scenarios requiring precise multimodal inference and responsiveness.

Similarly, **Qwen2.5-VL**, developed by Alibaba Cloud, significantly enhances open-vocabulary object localization, allowing operators to dynamically query for the detection and health state of infrastructure components simultaneously without predefined categories.

All tested models contain several billion (American billion) parameters, but a central design principle of the UAV Inspection Tool remains compatibility with the typical hardware constraints faced by Critical Infrastructure (CI) stakeholders. Most infrastructure operators do not have access to high-end GPU clusters, necessitating that the entire inspection solution run effectively on a single system with as little as 16 GB of vRAM. Given the computational intensity of modern VLMs, meeting this constraint poses a significant challenge. To ensure feasibility under these conditions, quantized versions of selected VLMs are deployed when necessary. This quantization approach significantly reduces memory usage while preserving strong inference performance, enabling stakeholders to leverage state-of-the-art AI capabilities without substantial hardware investments.

A detailed comparative analysis, as illustrated in Figure 19, confirms that these recently integrated models significantly outperform earlier generations such as LLaVA-1.6, even without relying on larger parameter counts.

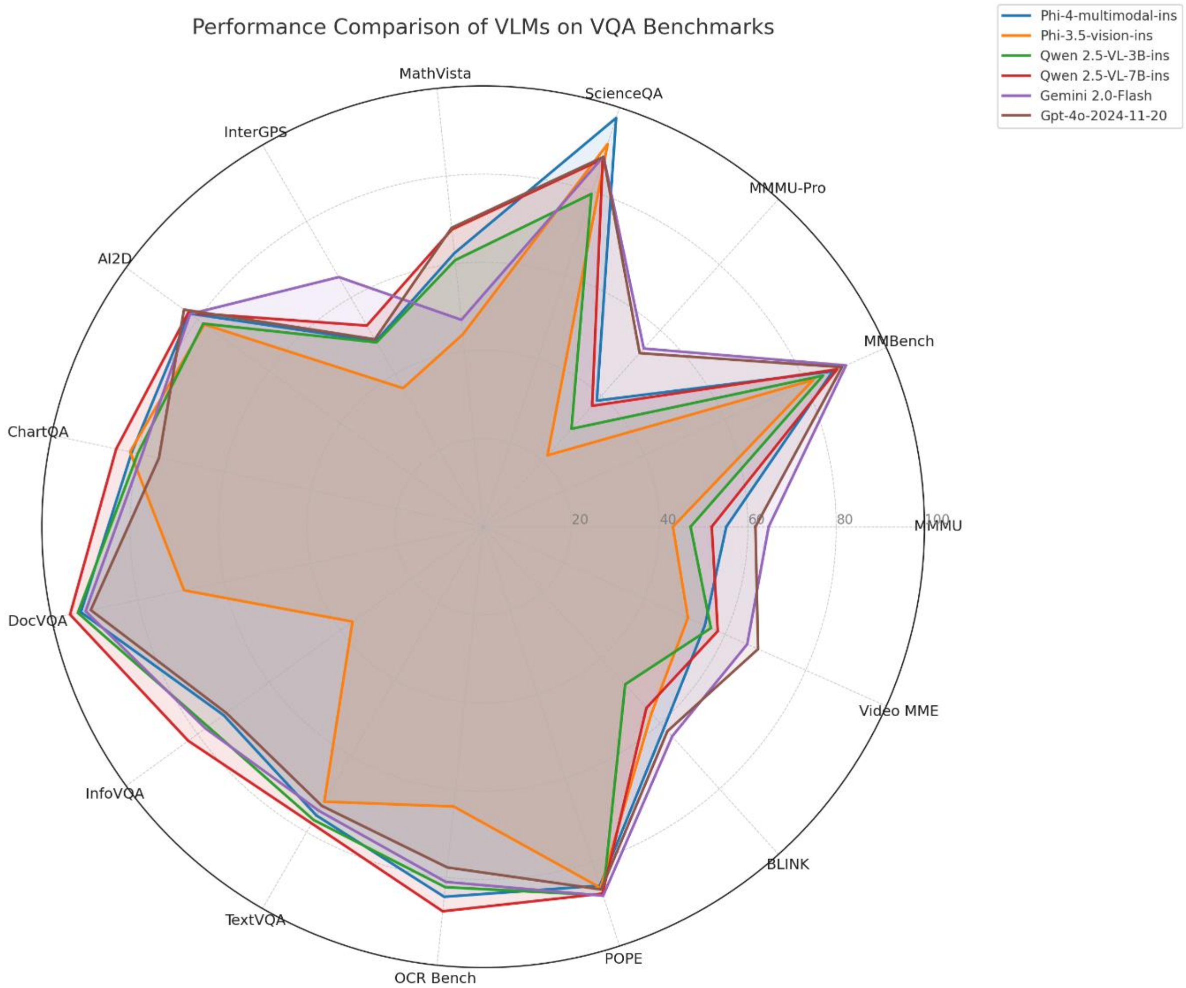

Figure 19. SOTA VLM for VQA. Data source: [56]

In challenging benchmarks like DocVQA, the Qwen VL 7B model achieved an accuracy of 95.7%, while the Phi-4 Multimodal-instruct reached 93.2%, both significantly surpassing LLaVA-NeXT's 85.7% accuracy. Similar improvements were observed in the MMBench benchmark, with Phi-4 Multimodal-instruct (86.7%) and Qwen VL 7B (87.8%) clearly outperforming previous-generation models averaging around 80% accuracy. Although Molmo-7B does not appear explicitly in Figure 19, its performance is comparable, with a DocVQA accuracy of 92.2% and an MMBench accuracy of 84.1%, underscoring its strong multimodal interpretative capabilities. These results underline the computational efficiency and interpretative power of the latest VLM integrations, ensuring superior performance in multimodal infrastructure assessment tasks.

CI Operators can interact seamlessly with this VQA module models through an intuitive graphical user interface (GUI), selecting inspection routines and submitting queries in natural language. The module promptly processes imagery and returns accurate textual responses, directly informing operational decision-making. By combining advanced AI-driven interpretive capabilities with straightforward user interaction, the system streamlines infrastructure inspection and maintenance processes, offering a scalable and adaptable solution for UAV-based monitoring. A comprehensive and detailed user guide can be found in Annex II (Annex II: UAV Image Processing API System User Guide) of this document, providing in-depth information on how this interaction between the operator and the VLM occurs.

This comparative analysis provides a clear view of the relative strengths and computational efficiencies, highlighting the real-world suitability of each integrated VLM for practical UAV-based infrastructure inspections.

As the dataset expands during Pilot 2, further validation will strengthen the solution's reliability and relevance for CI stakeholders. Ultimately, the true value of the UAV Inspection Tool lies not merely in the selection of any single VLM, but in its modular architecture, enabling seamless integration and replacement of models. This flexible approach ensures continuous improvement and adaptation to evolving operational scenarios and technological advancements, guaranteeing a balanced solution that meets both performance expectations and real-world deployment constraints.

3.3.3 3D Virtualization

The 3D virtualization module is designed to generate detailed three-dimensional reconstructions of infrastructure elements using UAV-captured footage. This capability enhances the inspection process by providing an interactive, spatial representation of structures, allowing operators to analyze them from multiple perspectives.

To achieve high-quality reconstructions, the system integrates multiple technologies that streamline the process from raw video input to a fully navigable 3D model. The core approach is based on **Neural Radiance Fields (NeRF) [53]**, specifically Instant-NeRF [54], which enables efficient and accurate transformation of 2D images into 3D scenes. **COLMAP [55]** is utilized for structure-from-motion calculations, extracting precise camera paths from UAV footage to ensure correct scene geometry. In parallel, detection and segmentation models such as GroundingDINO[23] and SAM-HQ[24] help isolate key infrastructure components, removing irrelevant background elements and improving the clarity of the final 3D model.

The workflow for generating 3D virtualizations follows these key steps:

- **Footage Collection**: UAVs record video from various angles, capturing the structure of interest comprehensively. A flight path that circles around the object or infrastructure of interest is highly recommended, as it provides a complete perspective and ensures optimal data for accurate 3D reconstruction.
- **Camera Path Estimation**: COLMAP processes the video to determine camera movement and spatial positioning.
- **Object Segmentation**: GroundingDINO and SAM-HQ filter out extraneous elements, focusing on the target infrastructure.
- **3D Model Training**: The processed images are fed into the NeRF model to create a fully interactive reconstruction.
- **Rendering and Output**: The generated model is visualized and exported for further analysis, allowing real-time navigation and measurement.

This modular pipeline ensures adaptability to different inspection scenarios, providing high-resolution 3D representations that facilitate infrastructure assessment.

While the module has demonstrated strong performance in controlled tests, several challenges remain. Processing times can be high, especially for large-scale reconstructions, limiting real-time applicability. File sizes of generated 3D models are often substantial, which can hinder smooth rendering and integration with lightweight systems. Additionally, the automation level of the workflow is still limited, requiring manual intervention in key steps such as segmentation and validation. Due to the low interest from Critical Infrastructure (CI) stakeholders and the prioritization of other key developments within the project, this module has not been a primary focus. Nevertheless, the foundational implementation is available, providing a functional solution for cases where detailed 3D visualization is valuable.

3.3.4 Anonymization

The anonymization module was developed to ensure compliance with data protection regulations, particularly **GDPR**, by preventing the collection of personal data during UAV-based inspections. The main concern was the potential capture of human faces in UAV images, which could raise privacy risks if stored or processed without safeguards. Since inspections can take place in public or semi-public areas, an automated anonymization step was necessary to ensure that only depersonalized data is handled within the system.

To achieve this, a deep learning-based detection system was implemented using **TinaFace [37]**, an AI model recognized for its real-time state-of-the-art accuracy in face detection tasks. A comprehensive review of existing models confirmed that TinaFace offers the best performance across major benchmarks, making it the optimal choice for this solution. Specifically, TinaFace achieved an Average Precision (AP) of 0.97 on the WIDER Face [39] dataset (easy), 0.963 on WIDER Face (medium), and 0.934 on WIDER Face (hard). Tinaface ResNet-50-based architecture provides high efficiency for real-time applications, ensuring robust detection under various lighting conditions, angles and occlusions. This capability makes it highly reliable for UAV-based inspections, where environmental factors can vary significantly. As shown in Figure 20, TinaFace ranks #1 in the WIDER Face benchmark, demonstrating its superior detection capabilities.

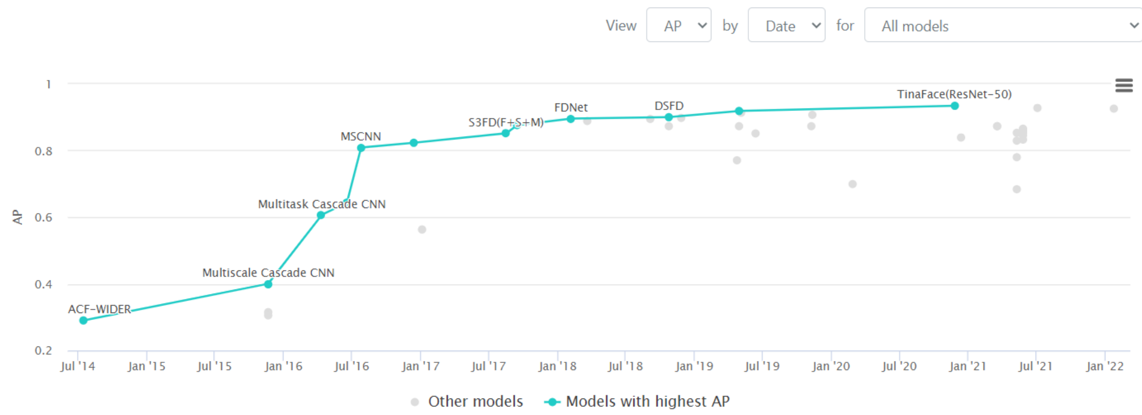

Figure 20: Face detection models leaderboard on WIDER Face (Hard). Source: [39].

The tool was deployed as an API service that automatically processes UAV-captured images before they enter the main AI inspection modules. If any faces are detected, they are blurred or black-boxed in real time before the image is stored or analyzed. This guarantees that all subsequent processing and potential storage of images occur only on anonymized data, effectively eliminating privacy risks.

Discussions with CI stakeholders explored the possibility of extending anonymization to license plates or full-body detection, particularly in areas where people might appear near inspected assets. However, field tests confirmed that such cases were rare and did not pose a significant privacy risk, making additional anonymization unnecessary. As a result, the tool remains focused exclusively on face anonymization, efficiently fulfilling its intended role without introducing unnecessary complexity into the processing pipeline.

3.4 Tool modules validation

The validation phase of the UAV-based inspection tool focused on assessing its **performance in real-world conditions** during the first pilot period at CI stakeholder facilities. This phase ensured that each module integrated effectively into the remote inspection workflow, confirming its practical applicability and identifying the strengths and potential limitations of the tool as a whole.

The evaluation covered final versions of the components, including object detection and segmentation, Visual Question Answering (VQA), 3D virtualization and anonymization. Each module was tested under operational conditions to verify accuracy, efficiency, and compliance with defined requirements. The results of these validations are also reflected in D7.4 Infrastructure Inspection Pilot Report[48], which provides further evidence of their effectiveness based on pilot data collected in real inspection scenarios.

As this deliverable represents the final complete version of the tool, it includes content demonstrating the functionality of all developed modules as an integrated solution, using captures extracted from the UAV web graphical user interface. Over the coming final months of the project, this validation will be further consolidated with additional data from Pilot Period 2, allowing for a more comprehensive assessment of the tool's operational impact.

Results are supported by real-world UAV data, with multiple demonstration videos available on the official YouTube channel. These include a first look at UAV-specific tool demos [38], real-time fire detection tests conducted at SZ facilities during Pilot 1 [49], and a general WP7 tool demo [50] showcasing integrated results within the work package's main interface.

3.4.1 Object Detection and Semantic Segmentation

This section presents the validation of the Object Detection and Semantic Segmentation module, highlighting its practical performance within the integrated UAV-based inspection solution during the Pilot 1 period. While initial quantitative evaluations have been previously conducted during model training, the focus here is on illustrating real-world applicability through qualitative results obtained from UAV inspections conducted at CI stakeholder facilities.

Examples provided in this section showcase the module's functionality in real inspection scenarios, demonstrating the system's capability to reliably detect structural anomalies such as cracks, corrosion and fire. These qualitative examples serve as a complement to the initial quantitative metrics available and provide a clear indication of the operational viability of the integrated solution.

First, it is worth highlighting the success achieved during the real-time detection tests conducted on live video streaming within the context of Pilot 1 at Slovenian Railways (SZ). During these tests, a controlled fire was deliberately ignited on the railway tracks to assess the operational effectiveness of the real-time detection module, specifically the YOLOv8-based fire detection model. Figure 21 provides examples of the results obtained, showcasing detections visualized in real-time and on-site during the test scenario. These results confirm the capability of the UAV inspection tool to accurately and swiftly detect real-time events, validating its operational readiness for real-world deployment.

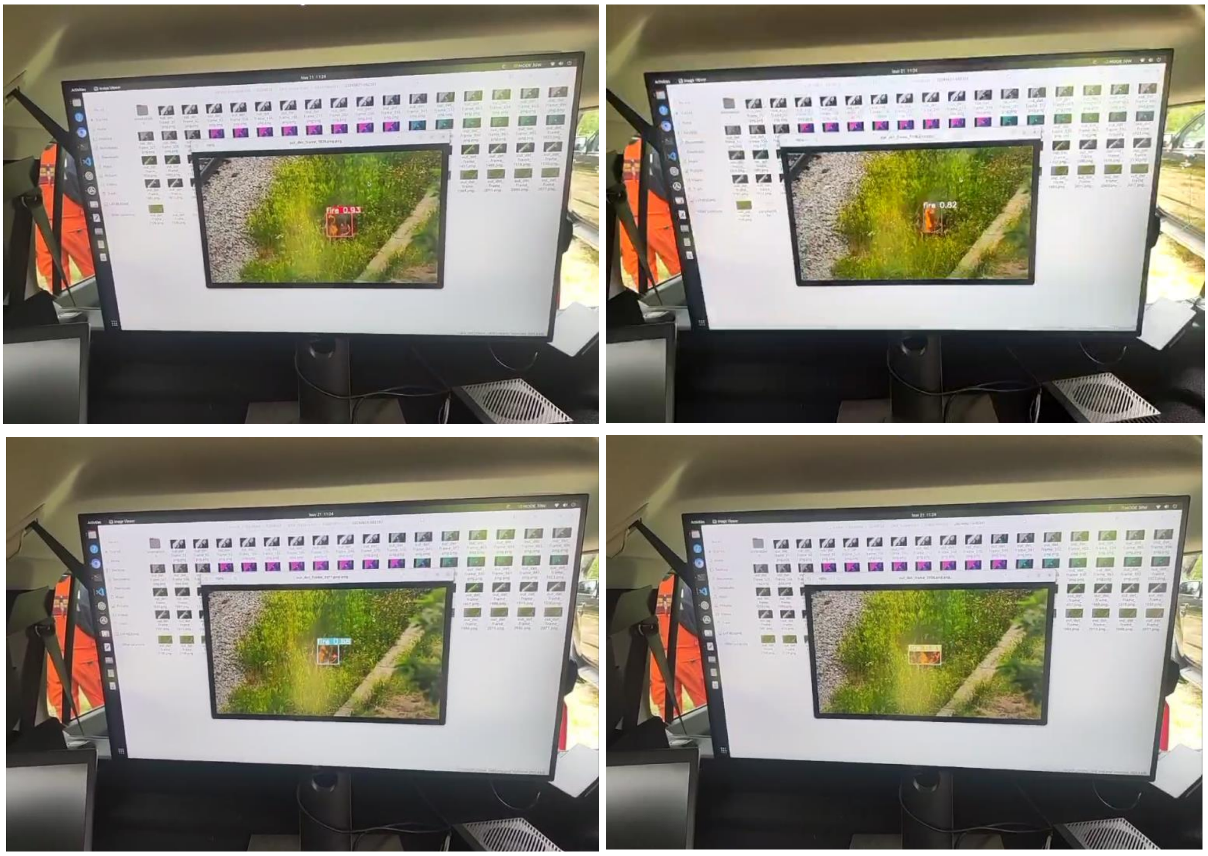

Figure 21: Examples of real-time fire detection UAV RII tool. Source: [48]. SZ facilities.

In addition to these real-time tests, in which the detection module is used as a standalone final detector rather than as the initial step in a more complex pipeline, further analysis was conducted using the footage collected during Pilot 1.

Figure 22 clearly exemplifies the combined capabilities of the detection and segmentation modules when used together with the VQA module. Firstly, the image demonstrates the effectiveness of the segmentation model GroundedSAM, which successfully segments pixels belonging to the class “concrete wall”. This significantly simplifies the image input for subsequent processing steps. Secondly, this clearer segmented image enables the crack detection module (YOLOv8) and subsequently the VQA module to perform precise analysis, accurately identifying and classifying structural conditions. This explicitly confirms anomalies like concrete damage.

This integrated approach allows robust performance by combining the strengths of each module. When all three modules confirm the anomaly detection, the system produces an accurate status vector of (1,1,1), automatically triggering the appropriate alarm and ensuring timely interventions based on reliable, actionable insights.

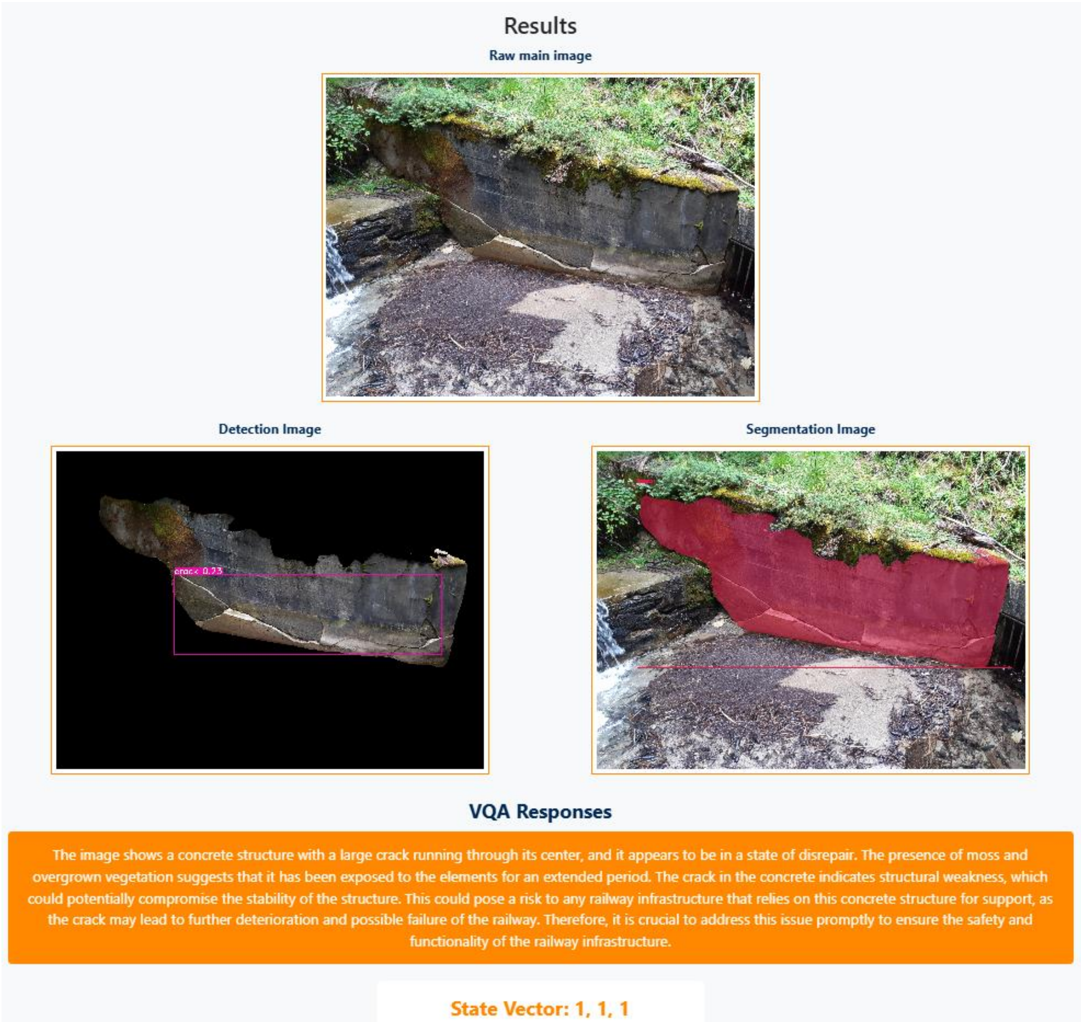

Figure 22: Crack detection example with UAV RII tool AI pipeline. HDE facilities.

While the previous examples demonstrate that ad-hoc trained detection models perform correctly (fire and cracks), the following two examples illustrate the significant advantage of using zero-shot open-vocabulary detection models. These models enable the identification and segmentation of objects of interest without the need to train or deploy specific models.

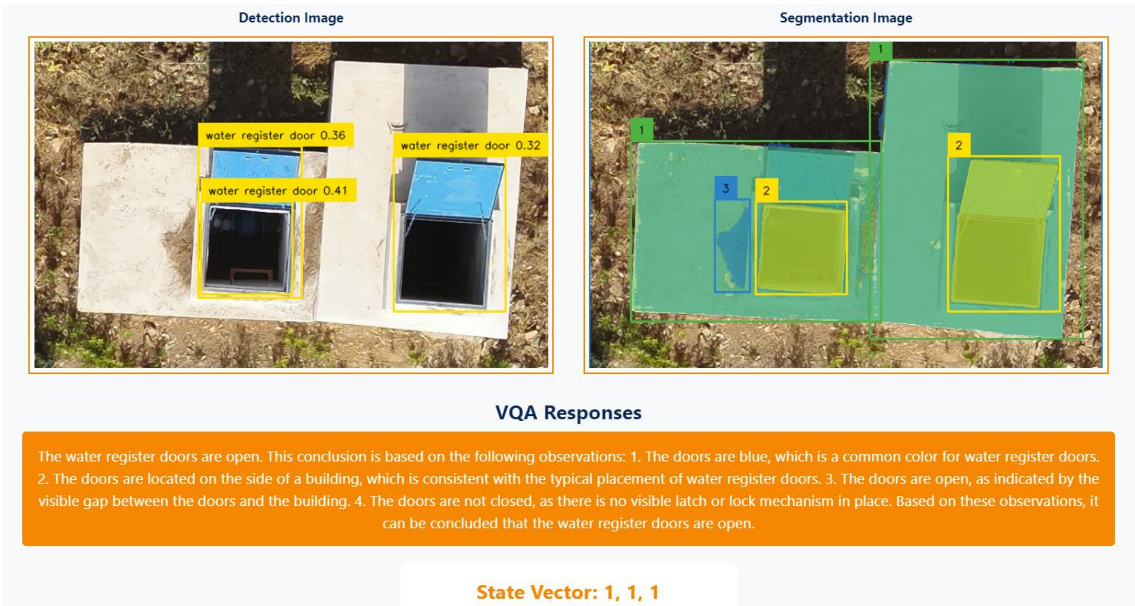

Figure 23: Open water register doors detection example with UAV RII tool AI pipeline. ACO facilities.

In Figure 23 (top), it is clearly visible that GroundedSAM successfully segments the classes "concrete structure," "water register door," and "debris" while also detecting the water register doors without issues. This enables the system to subsequently respond to questions regarding the status of these doors/trapdoors and notify the control center if they are open, as this could pose a safety risk.

In Figure 24 (bottom), the model similarly demonstrates its ability to detect and segment a previously undefined concept, such as a ceramic isolator. This output can then be used as an input to support the VQA module, which effectively understands and assesses the damage sustained by the device.

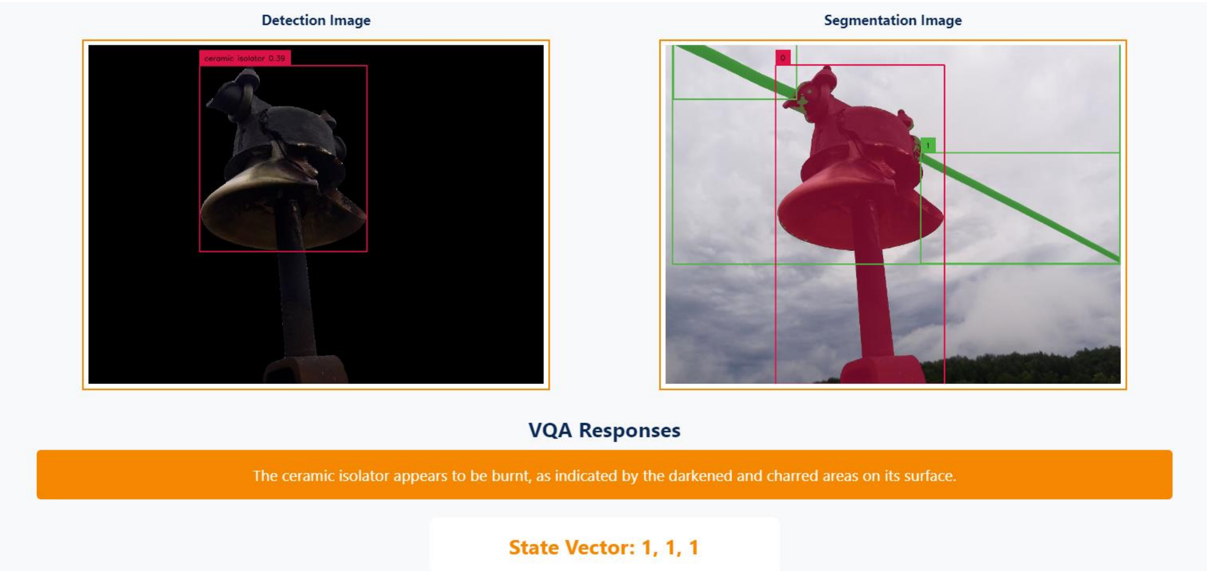

Figure 24: Broken ceramic isolator detection example with UAV RII tool AI pipeline. ELES facilities.

Another key feature worth emphasizing is the robustness provided by this composite solution. As illustrated in Figure 25, scenarios may arise where the initial segmentation performed by the open-vocabulary model GroundedSAM, using the classes "metal" and "corroded metal," yields good but improvable results. However, since the Rust YOLOv8 detection model and the VQA module operate in parallel, it is possible to triple-check whether corrosion is present in the dam infrastructure. This ensures that an alert can be triggered with high confidence.

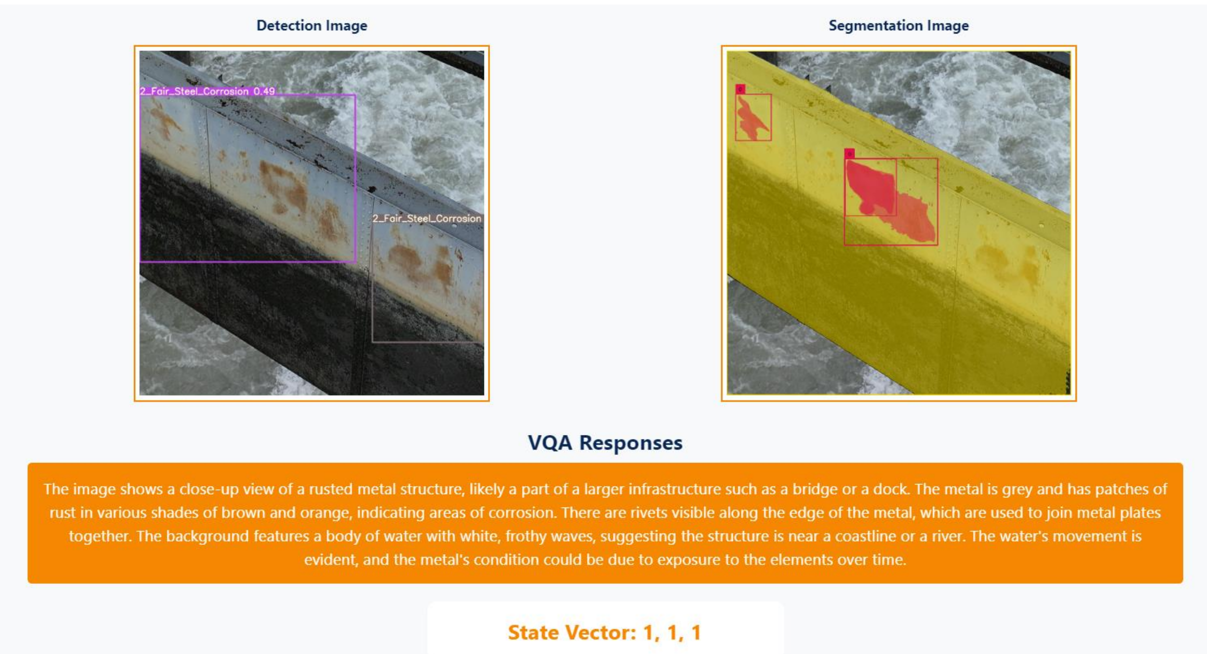

Figure 25: Dam rust detection example with UAV RII tool AI pipeline. HDE facilities.

Finally, an example image is included that was not captured during Pilot 1 but effectively demonstrates the capabilities of two additional models: YOLO-Worldv2 and X-Decoder. These models, combined with the text generated by the VQA module, clearly highlight the unusual nature of the scene, issuing alerts regarding both the obstruction of the track by a fallen tree and the flooding in the railway area.

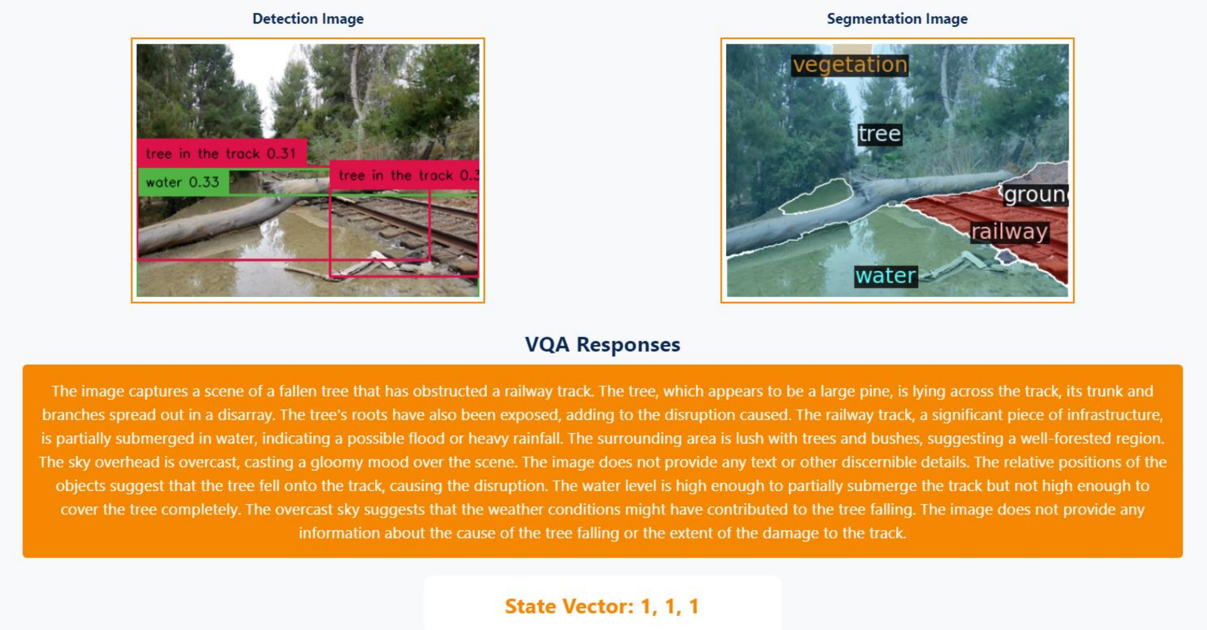

Figure 26: Flood and fallen tree detection example with UAV RII tool AI pipeline. Source: fullertonobserver.com

The diverse validation cases presented confirm the module's ability to operate effectively across multiple inspection challenges. The Pilot 1 results validate the effectiveness and operational viability of the Object Detection and Semantic Segmentation module within the UAV inspection tool. The combination of open-vocabulary models and task-specific detectors has demonstrated its ability to accurately identify and segment critical infrastructure anomalies, from structural cracks and corrosion to fire hazards and obstruction events. The integration with the VQA module further enhances the system's robustness, enabling multi-step verification and reducing false alarms. These findings reinforce the scalability and adaptability of the proposed approach, ensuring that the UAV inspection tool is well-equipped to support real-world infrastructure monitoring and decision-making processes.

3.4.2 VQA

In the previous section, valuable examples were presented that demonstrated the integration of the VQA module with the Object Detection and Semantic Segmentation tools, highlighting the system's ability to analyze UAV-captured imagery with a multi-step verification approach. However, in this section, the focus shifts towards a qualitative assessment of the VQA module itself, showcasing its ability to generate precise and context-aware responses based solely on visual input. These examples aim to illustrate the interpretative accuracy and contextual awareness of Visual Language Models across diverse inspection scenarios. Finally, the section will present quantitative evaluation results obtained through a small benchmark created using data collected during Pilot 1, comparing the performance of multiple VLMs in key infrastructure inspection tasks.

The first example highlights the remarkable adaptability of the proposed solution in response to natural disasters and emerging critical situations. Following the completion of the core developments of the UAV-based remote inspection tool, a catastrophic meteorological event known as DANA occurred in Valencia, Spain. This phenomenon resulted in heavy rains, river overflows and subsequent floods.

In response to this scenario, within a matter of minutes, a new pipeline was generated by leveraging the three integrated modules. The VQA module's descriptive capabilities were used to analyze the flooding status of city streets. An example of this analysis is shown in Figure 27, where the VQA response output states:

"The street appears to be partially submerged under floodwaters, indicating a significant waterlogging event."

Figure 27: Flood detection example with UAV RII tool AI pipeline in Valencia, Spain, after 2024 DANA. Source: efe.com.

This type of near real-time analysis, deployed in Valencia during the disaster, could have significantly improved early warning capabilities, enabling authorities to react sooner as the first floods and overflowing rivers emerged. Additionally, it could have provided valuable assistance during the post-disaster cleanup and recovery process, offering a centralized, real-time overview of street conditions. This adaptability to new and unexpected scenarios is one of the main strengths of the UAV inspection tool.

The second example further demonstrates the effectiveness of combining the three modules in a cascading pipeline. However, the key takeaway here is the VLM model's capability to go beyond simply identifying a cracked concrete sleeper. The model not only recognizes the defect but also interprets contextual markings, as shown in Figure 28:

"There is a yellow marking on the tie, which could indicate a specific maintenance or inspection note."

Additionally, since VLM models possess OCR (Optical Character Recognition) capabilities, a follow-up query could be submitted to extract the exact inspection note written on the sleeper, further enhancing the tool's functionality.

Figure 28: Railway concrete sleeper health status check example with UAV RII tool AI pipeline. SZ facilities.

The following three examples highlight the inspection of hydroelectric dam grates, focusing on clogged grates in remote and hard-to-access locations for HDE, our hydropower plant operator.

In the first case (Figure 29), the segmentation module effectively filters out irrelevant visual noise, ensuring that only the relevant grate area is analyzed in the next step. Thanks to this preprocessing, the VQA module accurately determines that the grate remains clear and in optimal condition, eliminating the need to trigger any alarm or maintenance action.

Figure 29: Clean grates detection example with UAV RII tool AI pipeline. HDE facilities.

The second and third images in this HDE use case depict the same location at different times of the year, Figure 30 in autumn and Figure 31 in winter. This comparison demonstrates the system's ability to track infrastructure conditions over time and detect seasonal issues, such as clogging caused by fallen leaves or snow accumulation.

In both cases, based on the VQA module's output, an alert is automatically sent to the WP7 General GUI, notifying the control center of the detected obstruction.

Figure 30: Autumn grates health status check example with UAV RII tool AI pipeline. HDE facilities.

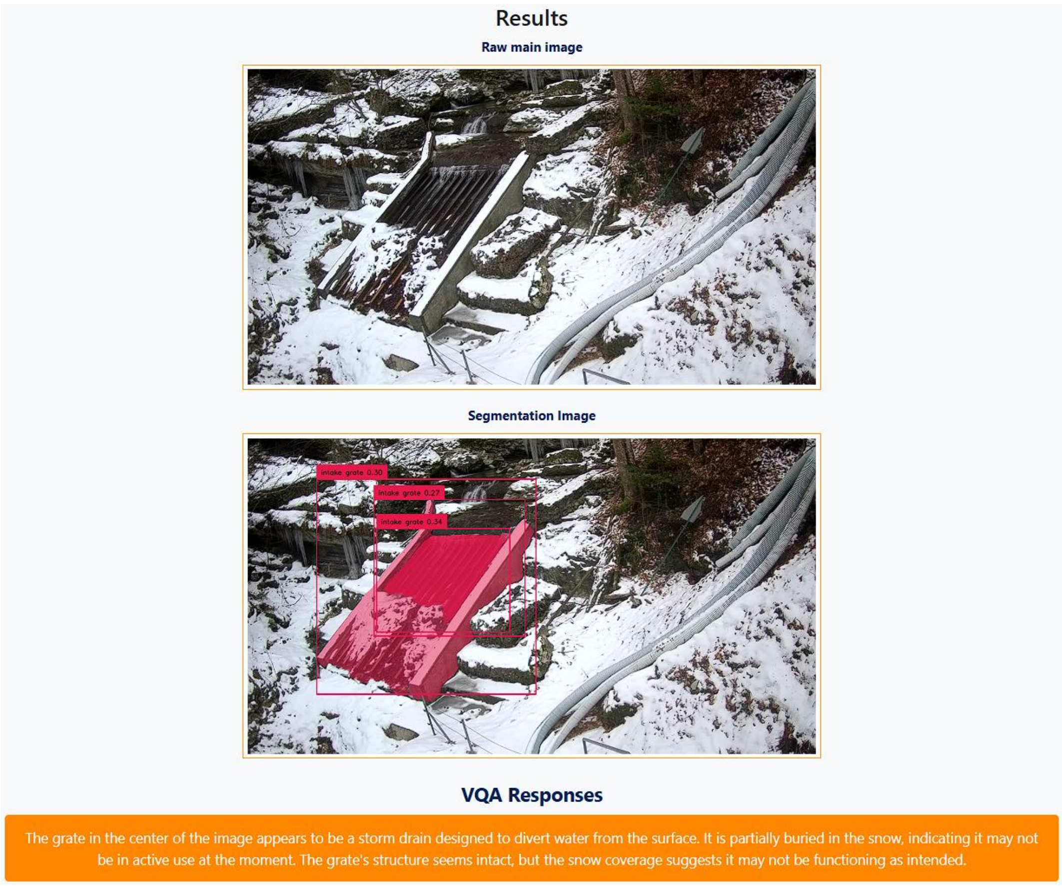

Figure 31: Winter grates health status check example with UAV RII tool AI pipeline. HDE facilities.

The final example showcasing the fully integrated inspection tool is presented in Figure 32. This case highlights the VQA module's ability to analyze multiple images within a single query, extracting valuable insights from both the visible spectrum and thermal imaging of a concrete ceiling in ACO's water tank facility in Málaga, southern Spain.

A key aspect of the VLM-generated response is the reasoning process followed by the model to reach its conclusions. The model correctly interprets the thermal and visual patterns, stating:

"The presence of discoloration and the heat distribution suggest that there may be leaks or other problems."

Figure 32: Multispectral image inspection example with UAV RII tool AI pipeline. ACO facilities.

After presenting this qualitative analysis, which highlights some of the key capabilities and strengths of the solution, the next step is to quantify the inspection results and assess their accuracy numerically. To achieve this, a small benchmark dataset was created, compiling representative images from all pilots, each corresponding to a relevant inspection use case.

Figure 33 provides an example of one of the approximately 50 entries included in this dataset. In this case, the output generated by the Molmo VLM is displayed along with its respective evaluation metrics. The associated JSON file contains essential information such as the image path (provided alongside the JSON for clarity), the user's query (stored under the "text" field), the dataset question ID, the human ground truth (representing the expected response defined by the CI stakeholders), the model's predicted response, and the extracted performance metrics.


```
"image": "/code/datasets/SUNRISE/images/vlcsnap-2024-10-31-16h35m56s431.png",
"text": "Is the wooden roof free of rockfall and rust? Answer yes or no and justify briefly your response.",
"question_id": 14,
"human_ground_truth": "No, many small rocks have fallen on the wooden roof.",
"Molmo-7B-D-0924": {
                "response": " No. The wooden roof is not free of rockfall and rust. The image shows a wooden roof covered in white rocks, indicating rockfall. Additionally, the wooden structure appears to be weathered, suggesting the presence of rust and other forms of deterioration.",
                "metrics": {
                    "accuracy": 1.0,
                    "f1_score": 1.0,
                    "bleu": 0.03515806390290379,
                    "semantic_similarity": 0.7326061129570007,
                    "exact_match": 0.0,
                    "llm_eval": 1.0,
                    "llm_eval_explained": "answer: yes. both sentences convey the notion that there has been some form of damage (rockfall) affecting the wooden roof. while one mentions \"many\" small rocks, the second describes it more explicitly stating an image showing the effect (\"covered in white rocks\"). however, both convey the general concept that something went wrong with the wooden roof due to external factors like natural elements falling onto it.",
                    "rouge1": 0.19607843137254902,
                    "rougeL": 0.15686274509803924
                }
```


Figure 33: Example UAV SUNRISE benchmark entry. VQA results and metrics. HDE facilities.

By applying this evaluation process across all dataset entries and available VQA models, the results summarized in Table 3 were obtained.

It is important to note that VLM technology evolves at an exceptionally fast pace, with new models continuously emerging that outperform their predecessors on a weekly or, at most, monthly basis. For the latest version of the inspection tool, Phi-3.5 Vision has been selected as the default VLM, striking an optimal balance between performance and hardware requirements. However, this extended benchmark analysis underscores that the true added value of the solution lies not in a single model but in the combination of multiple models, the seamless integration of new architectures and the flexibility of the overall approach.

Table 3. Benchmark results on pilot 1 data.

Model	Binary Accuracy (%)	Content Accuracy (%)	Model size (parameters)
Phi-3.5-vision	74.07	74.07	4.2 B
Phi-4-multimodal-instruct	78.43	74.07	5.6 B
Qwen2.5-VL	79.63	79.63	7 B
Molmo	85.19	83.33	7 B

As can be inferred from the table above, at the time of publication of this deliverable, the results obtained with the small collected dataset show that Molmo is the model with the highest performance for the VQA task in critical infrastructure inspection. It achieved an accuracy of 85.19% when the response is evaluated in binary terms (alarm triggered or not), and 83.33% when considering whether the content of the response and its justification align with the ground truth defined by the human operator. Despite these results, Phi3.5-Vision is considered to offer a better balance between performance and hardware requirements (4.2 billion parameters compared to Molmo's 7 billion), making it the most suitable option for deployment in the tool.

Thus, while the dataset is expected to expand further during Pilot 2, the results already confirm the tool's effectiveness across key infrastructure inspection use cases, validating the integrated solution approach adopted in WP7.

3.4.3 3D Virtualization

To evaluate the practical applicability of the 3D virtualization module, a real-world test was conducted using UAV footage of a dam facility managed by HDE. This case study aimed to determine how effectively the module could reconstruct a complex infrastructure element and support remote inspection workflows by enabling spatial analysis and virtual exploration.

Following the methodology described in Section 3.3.3, a Point of Interest (POI) flight path was executed around the dam, capturing high-resolution footage from multiple angles to provide a complete dataset for reconstruction. The COLMAP tool was used to estimate the UAV's trajectory and camera positioning, while GroundingDINO and SAM-HQ segmented the structure, isolating it from irrelevant background elements. The processed frames were then input into Instant-NeRF, generating a high-fidelity 3D model of the dam that could be navigated interactively. This process is illustrated in Figure 34, and can be seen in more detail in the official **YouTube channel video [38]**, starting at minute 2:50, where the real-time rendering of the model for dynamic input angles is demonstrated for this case study and another example.

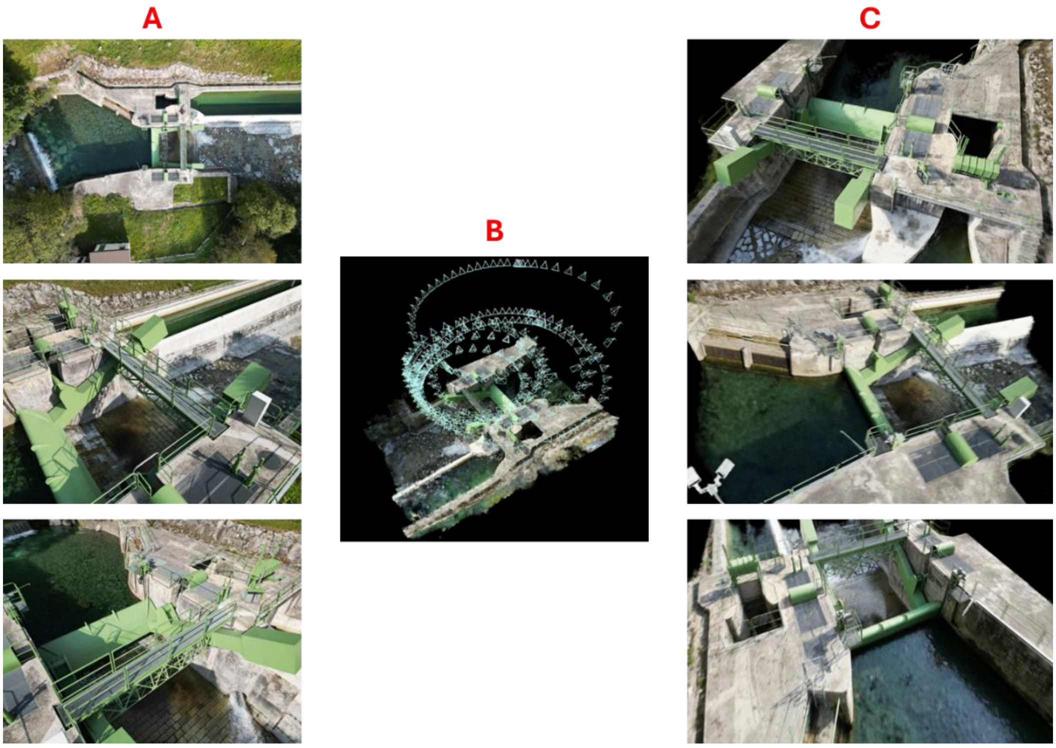

Figure 34: 3D virtualization HDE's dam reconstruction. (A) images extracted from raw POI videos; (B) camera path reconstruction; (C) output 3D rendering video frames. Source: [47].

The generated 3D mesh (exported in ".obj" format), shown in Figure 35, enabled spatial measurements and remote inspection of structural components. To ensure that the measurements taken from the virtual model corresponded to real-world values, a conversion factor was required. This factor is calculated by selecting a known reference dimension from the real-world structure, which serves as a scale for all other measurements within the 3D environment.

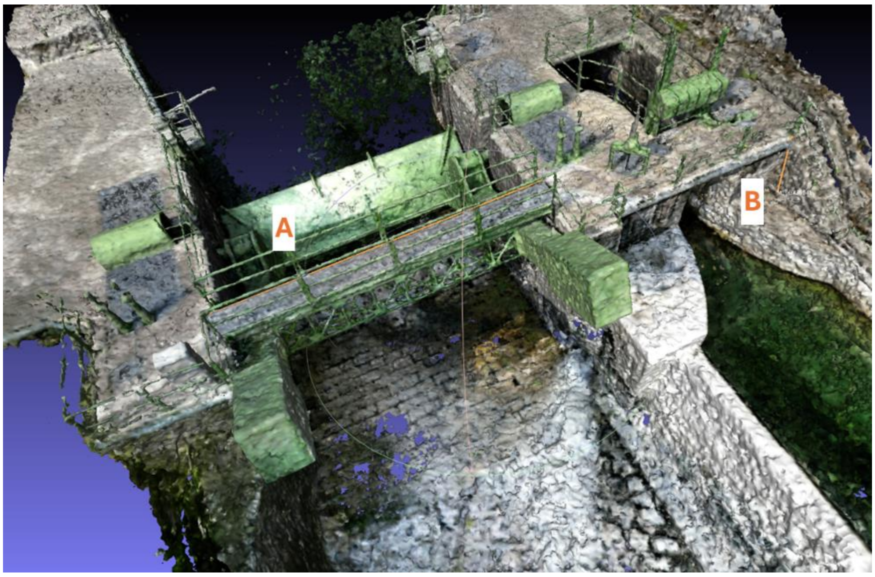

Figure 35: HDE dam 3D mesh measurement estimation. Orange line (A) represents bridge longitude measurement; Orange line (B) represents stair height measurement. Source: [47].

For this case study, the dam bridge length (A) at 8 meters and the stair height (B) at 1.75 meters were used as reference points to establish a scale conversion factor. By comparing these real-world dimensions with their corresponding raw measurements in the 3D mesh, conversion factors of 3.52 for the bridge and 3.76 for the stairs were calculated, as shown in Table 4's data. The consistency between these values confirms the reliability of the reconstruction for spatial analysis.

Table 4: 3D mesh distance measurements. GT vs 3D model distance before scale factor application.

Measured concept	Ground truth (m)	Raw 3D mesh
(A) Bridge longitude	8.00	2.270
(B) Stair height	1.75	0.465

Applying these conversion factors to additional model-derived measurements resulted in an average deviation of 5% from real-world dimensions. This discrepancy is primarily attributed to manual annotation imprecision when selecting measurement points within the 3D environment, as well as potential reconstruction errors inherent to the processing pipeline. Despite this margin of error, the case study successfully demonstrated the practical utility of 3D virtualization as a complementary functionality within the UAV inspection workflow, offering an alternative method for remote assessment in scenarios where direct access is restricted.

However, the large file sizes of the resulting 3D mesh (~32MB in this case) present challenges for seamless web interface visualization. Given these constraints and the prioritization of other critical project developments, further refinements to this module were not pursued. Nonetheless, this validation using real-world data confirms that 3D reconstruction provides a viable enhancement to infrastructure inspections, particularly when detailed spatial analysis is required.

3.4.4 Anonymization

To validate the effectiveness of the anonymization module, a series of tests were conducted using UAV-captured images containing human figures. The goal was to confirm that the tool reliably detects and anonymizes faces before images enter the AI inspection modules, ensuring compliance with GDPR and other data protection regulations.

The anonymization process was tested using images captured at CI stakeholder facilities where controlled test conditions allowed planned human presence. The system, powered by the TinaFace AI model, successfully identified and obscured faces in real-time, ensuring that no personal data was stored or processed. Operating as an automated API service, the tool applies face detection and anonymization (blurring or black-boxing) before any UAV image is analyzed, as illustrated in Figure 36.

Figure 36: Anonymized images taken in ACO and HDE facilities. Simple and complex scenarios, close-up, partially occluded, far away, or rotated faces. Source: [47].

The system was tested under diverse conditions, as shown in Figure 36, before its deployment in the first pilot period, where its effectiveness was confirmed in real-world operations. Throughout these pilots, the anonymization module ensured that no images containing personal data were stored at any stage of the inspection process. The tool functioned seamlessly, processing all captured images and applying face anonymization before storage, effectively demonstrating its compliance with data protection requirements in operational scenarios.

These results confirmed that the anonymization tool meets the required privacy standards while maintaining efficiency in operational environments. The system maintained consistent performance across different lighting conditions, viewing angles and image qualities, while its real-time processing capability ensured that anonymization did not introduce significant delays into the workflow. This seamless integration allowed UAV-based inspections to remain both efficient and fully compliant with privacy regulations.

3.5 Deployment

Building on the architecture and operational principles described in Section 3.2 and incorporating practical insights gained from Pilot Period 1 (extensively detailed in D7.4 Infrastructure Inspection Pilot Report [48]), the UAV Inspection Tool is consistently deployed as a **REST API-based service**. The flexibility of this approach allows these APIs to be hosted either on a dedicated edge device—specifically, the Jetson Orin AGX 32GB system—or on centralized servers or cloud environments, depending on operational requirements and site-specific constraints.

During the first pilot deployments, two primary hardware configurations were validated:

- **Edge-based deployment**, utilizing the Jetson Orin AGX device.
- **Centralized deployment**, running on a remote laptop equipped with a dedicated 16 GB vRAM GPU.

Regarding **edge-based deployment**, although the architecture supports mounting the Jetson device directly onboard the UAV, operational considerations during Pilot Period 1 favored an intermediate solution. Specifically, UAV-acquired imagery is transmitted in real-time via RTSP streaming to the Jetson device located at the UAV control station, enabling edge processing without risking damage to critical computational equipment due to onboard vibrations, battery limitations, or environmental factors.

The images in Figure 37 illustrate the deployment setup used in the Pilot 1 tests to evaluate the real-time fire detection solution. A detailed demonstration can be seen in this video: [49].

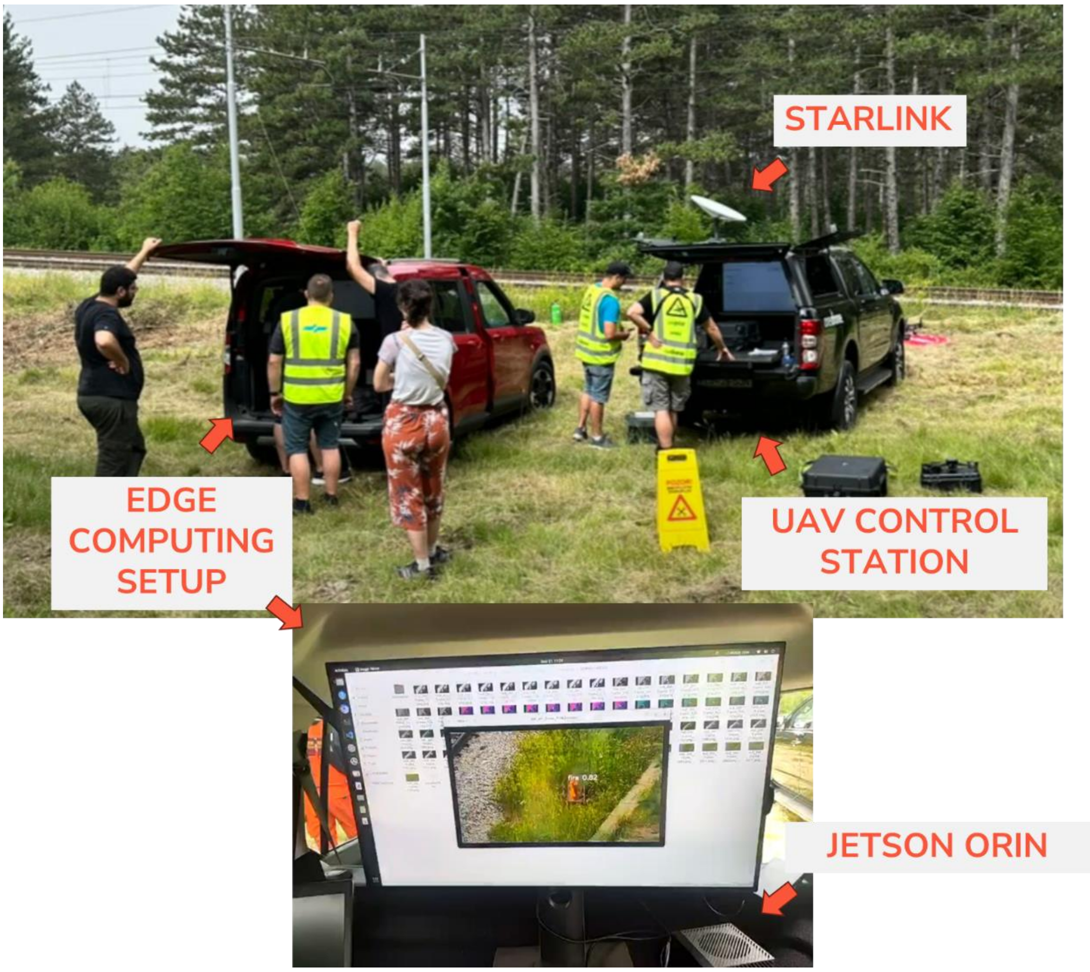

Figure 37: Set-up for real-time solution deploy on Pilot 1. SZ railway facilities.

The plans for the final Pilot 2 are also oriented towards avoiding mounting the Jetson directly on the UAV, as long as real-time streaming to the edge computing device can be ensured. To achieve this, a Starlink connection will be available to mitigate the limited bandwidth issues encountered during the initial field tests. In Pilot 2, Starlink will be used exclusively for real-time AI video analysis, ensuring stable and high-speed connectivity whenever real-time processing is required. It serves as a hotspot to connect the Jetson Orin to the same network where the UAV video stream is transmitted. This setup was tested during Pilot 1, where previous bandwidth limitations were identified while using a 4G mobile hotspot, demonstrating the need for a more robust and dedicated connection.

Regarding **centralized deployment**, the primary modification deployed in this mode of operation has been the implementation of a REST API for exchanging image or video files, replacing the previous MQTT broker-based transmission using Base64 encoding. This API is hosted on the same local server as the WP7 monitoring GUI, which centralizes alarm management. This change was introduced during Pilot 1 to resolve latency and bottleneck issues observed in the initial field tests, significantly improving data transfer efficiency, reducing system load, and enhancing stability. The API is already fully integrated and will continue to be used in Pilot 2. Additionally, to ensure secure, reliable, and low-latency communication between all system components—including the UAV control station, Jetson devices, centralized servers, and WP7 GUI—a ZeroTier private VPN has been implemented. This VPN ensures encrypted and stable connectivity without additional manual authentication or configuration, contributing significantly to the ease of deployment and integration.

In both deployment scenarios, the process has been simplified and standardized through **Docker containerization**, ensuring consistent and reliable operation across varied computing environments. Software management and updates are centralized via a **GitHub repository**, enabling seamless synchronization, version control, and simplified deployment procedures without operational disruptions.

A significant effort during this final integration phase has been dedicated to the development and refinement of the **UAV-specific GUI**, addressing the challenges faced by end-users in autonomously processing images. As a result, in this final version, operators can manage inspections through a dedicated UAV-specific web interface. As emphasized by users who tested the interface, this GUI provides an intuitive platform for initiating inspections, selecting predefined inspection routines, and configuring custom parameters according to the specific scenario. Users particularly highlighted its ease of use, clear visualization of inspection results, and rapid adoption without extensive training. Additionally, seamless synchronization with the main WP7 monitoring interface ensures efficient operational workflows.

Once data acquisition is complete, detailed inspection results are visualized directly within the UAV interface, offering immediate insights into detected anomalies and actionable information. Additionally, critical alerts and high-level inspection summaries are fully synchronized with the general WP7 monitoring interface, ensuring a comprehensive and centralized overview of infrastructure health and inspection activities.

All message exchanges and alert publication between the UAV GUI and the WP7 GUI have been successfully tested and validated, confirming seamless integration. Further details about the WP7 GUI functionalities and its role in the inspection workflow can be found in Section 4 of this document.

For a detailed breakdown of the UAV GUI, refer to Annex II of this document. As a quick overview, a demonstration video [57] has been prepared to guide CI operators through the GUI workflow using a specific use case. Figure 38 illustrates the step-by-step process explained in the video, showcasing how the UAV GUI integrates with the main WP7 alert management interface.

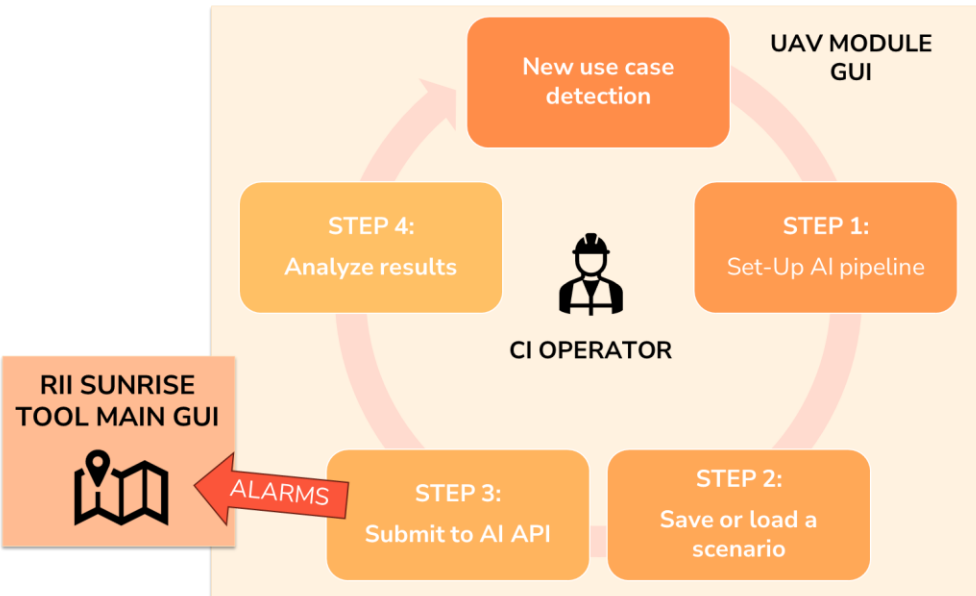

Figure 38: UAV GUI new inspection use case configuration. Video demo **[57]**.

In conclusion, the deployment strategies validated during Pilot Period 1 have confirmed both the edge and cloud-based REST API hosting options as effective and practical. Moving into Pilot Period 2, further consolidation of deployment approaches and operational procedures will be undertaken, with continued evaluation of the potential need for direct UAV hardware integration, contingent upon evolving inspection demands and stakeholder feedback.

3.6 UAV platform integration

During the first pilot, we achieved promising results by focusing on the UAV inspection ability, operating within Visual Line of Sight (VLOS) to ensure controlled and effective testing. For the second pilot, our objective has expanded to evaluate the drone solution as a comprehensive system. To support this, we have implemented a new Unmanned Aerial System (UAS) specifically designed for Beyond Visual Line of Sight (BVLOS) operations.

For Pilot 2, we have developed a new, robust UAV platform designed specifically for Beyond Visual Line of Sight (BVLOS) operations to tackle the demanding flights required at the HDE premises in the Italian Dolomites, as well as the SZ and ELES premises in Slovenia. While the focus in previous phases was on testing the capabilities of the data collected during flights, this phase shifts towards evaluating the entire solution. This includes assessing the drone's endurance, its performance in BVLOS operations, and its overall reliability. The goal is to demonstrate the strengths of the system to critical infrastructure (CI) stakeholders, enabling them to evaluate its potential for adoption and integration into their operations.

3.6.1 Aerial Vehicle Hardware Specifications

In this phase, we will deploy the T-drones MX860 UAS as depicted in Figure 39, equipped with a Starlink mini satellite dish to ensure a robust and stable Command and Control (C2) link. This setup is critical for successful operations in the mountainous regions of the critical infrastructure (CI) stakeholders. Our focus is not only on testing the inspection algorithms but also on assessing the drone's performance and the communication system's endurance under challenging conditions.

The T-drones MX860 is a professional-grade Unmanned Aerial System (UAS) specifically chosen to meet the primary requirement of flying 5 kilometers and returning within approximately 40 minutes. This capability was a key factor in selecting the appropriate UAS for the demanding operations required in Pilot 2. The system is designed to deliver safe, reliable and efficient performance while addressing the challenges of advanced operations with a strong emphasis on safety, endurance and operational control.

The UAS is equipped with advanced position-keeping capabilities in 4D space (latitude, longitude, altitude, and time), enabling precise navigation and safe operation near obstacles, even at distances closer than 30 meters. It continuously monitors critical flight parameters, including position, altitude, speed, attitude, trajectory and energy status, ensuring comprehensive situational awareness during missions.

The system integrates state-of-the-art navigation and obstacle avoidance technologies, allowing for both pre-programmed flight paths and dynamic route adjustments while maintaining containment within defined operational volumes. Its command and control (C2) links are protected with interference mitigation mechanisms and advanced security features to prevent unauthorized access. Additionally, the UAS is equipped with reliable fail-safe mechanisms to recover from C2 link losses or safely terminate flights when necessary.

From a safety perspective, the T-drones MX860 incorporates both basic and enhanced containment measures to ensure operations remain within authorized areas. The human-machine interface is designed for clarity and efficiency, reducing operator fatigue and minimizing the risk of errors. The system also includes modern remote identification capabilities and is equipped with lighting for night operations when required.

In summary, the T-drones MX860 is a highly capable and reliable UAS, designed to deliver safe and efficient performance in a variety of operational scenarios. Its advanced features, robust safety measures, and ability to meet the endurance and range requirements make it an ideal solution for the challenging missions of Pilot 2.

Figure 39: Aerial Vehicle T-drones MX860.

Table 5. UAV Specs.

UAV Specifications	Performance Metrics
Maximum Flight Speed	20 m/s
Maximum Flight time (incl. payload, battery)	Approximately 40 min
Maximum Wind Resistance	14 m/s
Type	Coaxial 8-rotor
Minimum number of operators	1 operator
Landing Equipment	Carbon Landing Skids
Arms	x4 foldable/ unfoldable motor arms
Dimensions (folded, without propellers)	433mm x 413.5mm x 250mm
Length with propellers	1167mm.
Width with propellers	1167mm.
Maximum Take Off Weight (MTOW)	19.96kg
Fuselage Ingression Rate	Designed for IP52
Operational Temperature	-15° to +50° C
Relative Humidity (ground)	0 to 80% (without condensation)

Figure 40: Aerial Vehicle T-drones MX860

Figure 41: Aerial Vehicle T-drones MX860 with camera payload

3.6.2 UAS Connections and Communications

The connections and communications have been finalized, and the connectivity diagram is presented in Figure 42. The updates from the previous deliverable, D7.3 [46], include the use of the Starlink Mini mounted on top of the drone to ensure robust command and control (C2).

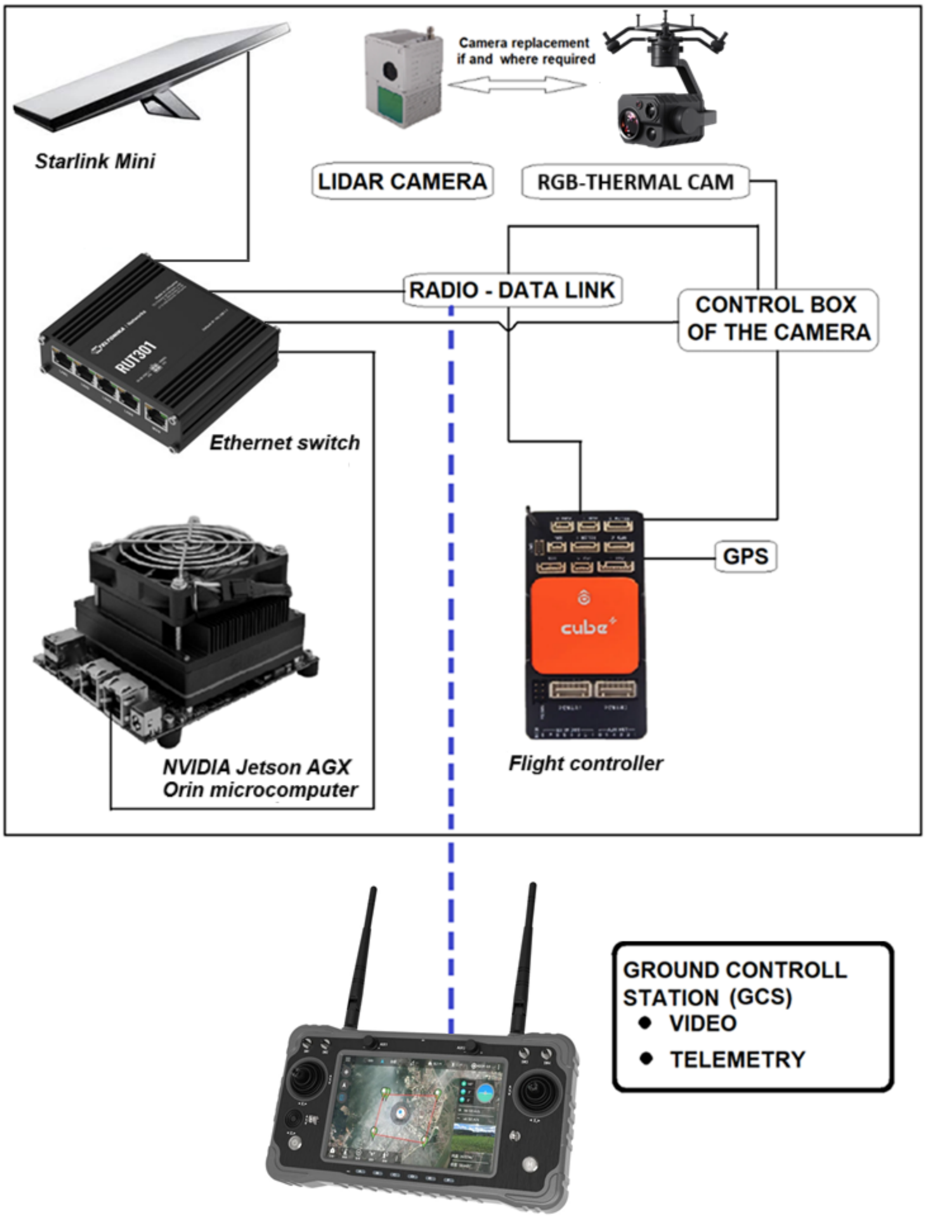

Figure 42: Inspection Tool: “UAS Connection Diagram and Communication”.

3.6.3 Relay Drone System

One of the key advancements in Pilot 2 is the adoption of the Starlink Mini for BVLOS operations in mountainous regions, replacing the initial plan to use a relay drone. This modification ensures compliance with EASA regulations for BVLOS flights by providing the most secure and reliable method for maintaining a direct Command and Control (C2) link. By enhancing operational safety and integrating emerging technologies, this approach demonstrates an innovative and forward-thinking solution to the challenges of operating in remote and rugged environments.

4 User interface for remote infrastructure inspection

This chapter provides the final description of the components of the User Interface dashboard dedicated to inspecting critical infrastructure events captured from UAVs or Satellites, continuing the work from Task T7.2.

4.1 General context

The content of this chapter remains consistent with the previous deliverable. The Dashboard User Interface (UI) continues to function as a critical component for monitoring and analyzing infrastructure inspection data collected from UAV and satellite inspection abilities, preserving its established design, structure, and functionality.

The design and architecture of the Dashboard UI remain unchanged, with its primary objective being the delivery of real-time images to users. These images highlight areas, components, or points of failure in critical infrastructure, such as damaged components, structural issues, corrosion, and vegetation obstructions.

The Backend Coordinator, the core component of the solution, operates seamlessly, ensuring smooth integration and functionality across the system.

4.2 Architecture: High level Implementation

To provide a better understanding, it is necessary to revisit the architectural framework behind the development of the Dashboard UI. This will help readers easily interpret the content shown in Figure 43. It is thoroughly detailed in deliverable D7.2[46], section 4.2. The web application architecture through the numbered bus lines as follows:

1. Incoming messages/events from UAV/Satellite systems are received. All this data is routed through an MQTT bus system. Within this system, the data is systematically queued, ensuring a sequential flow.
2. The Backend Coordinator processes all incoming messages/events. It retrieves the data at the front of the MQTT queue.
3. All incoming messages/events are internally stored in the Backend Inventory (MongoDB server).
4. The Backend Coordinator sends live or historical data to the Dashboard UI for visualization and responds to historical data requests from the Dashboard UI.
5. The Dashboard UI communicates with the Google Maps infrastructure to render maps, markers, points of interest, and heat maps, among other elements.
6. The Backend Coordinator sends requests to the Reporting Subsystem in order to compile the requested data and then receives the results.
7. The Reporting Subsystem and the Backend Inventory communicate with each other in order to process the requests and subsequently transmits the results to the Backend.
8. The Dashboard UI obtains an Access Token from the Identity Server to access backend APIs. Access to the UI is exclusively granted to authorized users, with authentication and authorization handled by a dedicated Authentication/Authorization unit, responsible for controlling user access and logging into the application.
9. Additional public services can offer crucial meteorological data, weather forecasts, maritime information, alerts, and more for visualization within the Dashboard UI.

It is important to note that the connection between the Backend Coordinator and the MQTT system is bidirectional. If any data needs to be transmitted from the application outward, the Backend Coordinator places it in MQTT, within the corresponding queue.

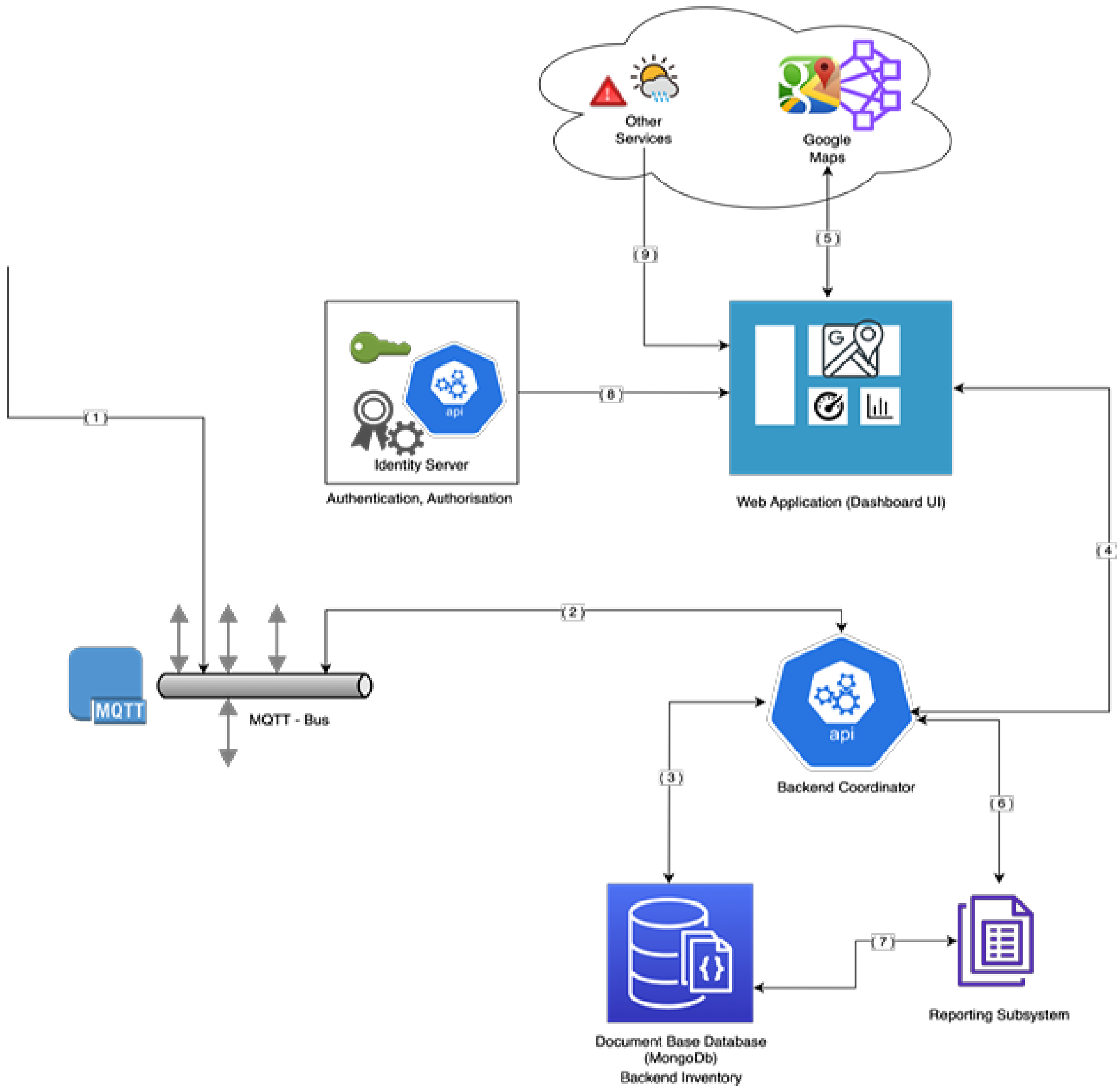

Figure 43: UI Architecture Diagram.

4.2.1 Internal Components - Backend Coordinator

The Backend Coordinator is composed of several key background services that ensure seamless data processing, storage, and interaction between system components. These services work together to manage event data, facilitate real-time communication, and support user-driven reporting and visualization. Below is an overview of the primary services and their respective roles within the system.

MongoDbService. This background hosting service is responsible for writing all events received from the corresponding Detection Services to the "UAV" and "SAT" collections.

MqttSubscriberService. This background hosting service is responsible for listening to the MQTT topics and forwarding these messages to the Dashboard UI using WebSocket communication for real-time event presentation. It utilizes the MongoDbService to store these messages in the Document-Based Database (MongoDB) Backend Inventory.

ReportingService. This background hosting service is tasked with generating filters based on criteria selected by end users on the Dashboard. It listens to user requests from the Dashboard for searching and reporting purposes. Upon receiving these requests, it executes queries on the Document-Based Database (MongoDB) Backend Inventory through the MongoDbService of the Backend Coordinator Service. Subsequently, it returns the results to the Dashboard Web App for further visualization and exporting functionalities.

4.2.2 Database security

The MongoDB implementation within the system is designed with a strong emphasis on security to ensure the integrity, confidentiality, and availability of data. Exclusively accessed by the Backend Coordinator service, the database is fortified with advanced security measures, including robust authentication, encryption, auditing, and network protection. This section outlines the key security features and best practices implemented to safeguard the MongoDB environment and maintain its resilience against potential threats.

Authentication and Authorization: The Backend Coordinator service serves as the solitary user of our MongoDB databases, managing authentication and authorization processes. Authentication is enforced through a robust username/password mechanism, allowing the Backend Coordinator service to securely access MongoDB resources. Role-based access control (RBAC) is meticulously configured to grant the Backend Coordinator service granular permissions, restricting its access to databases and operations based on predefined roles.

Encryption: Our MongoDB deployment, exclusively accessed by the Backend Coordinator service, implements encryption at rest and in transit to fortify data security. Data at rest is safeguarded using the WiredTiger encryption engine, employing AES-256 encryption to protect data files stored on disk. Encryption in transit, enforced through TLS/SSL protocols, ensures that data exchanged between the Backend Coordinator service and MongoDB servers remains encrypted during transmission, safeguarding it from unauthorized access and tampering.

Auditing and Logging: The Backend Coordinator service integrates robust auditing and logging functionalities, enabling comprehensive tracking and monitoring of all database activities. Audit logs meticulously record authentication attempts, database commands, and administrative actions initiated by the Backend Coordinator service, facilitating security analysis and compliance auditing.

Network Security: Stringent network access controls are configured by the Backend Coordinator service to restrict access to MongoDB servers exclusively to authorized entities. Network encryption using TLS/SSL protocols is enforced to secure data transmission between the Backend Coordinator service and MongoDB servers, bolstering network security and mitigating the risk of unauthorized access.

Authentication Plugins: As the sole user of MongoDB, the Backend Coordinator service does not require integration with external authentication systems. Authentication is exclusively managed through the username/password mechanism.

Security Best Practices: Our MongoDB deployment, managed by the Backend Coordinator service, adheres to industry-standard security best practices. Secure deployment configurations, access control policies, encryption settings, and auditing configurations are implemented to uphold the integrity and security of our MongoDB environment. Regular updates and patching are diligently performed to address security vulnerabilities and maintain the resilience of our MongoDB infrastructure.

4.2.3 Rest API protocol of Dashboard UI

The security of critical infrastructures is a top priority that requires our utmost attention. Given the importance of this topic, it is necessary to revisit the design considerations from the previous deliverable, D7.2[46]. This will ensure that the security aspects are thoroughly addressed and incorporated into the overall system design.

The Authentication, Authorisation and Audit Logging component is responsible for intelligently controlling access to UI tools' system functions and interfaces (both GUI and REST-API), enforcing policies and keeping an audit trail of events happening. Based on assigned roles, authenticated users can access different UI system functions and interfaces. The audit logging mechanism can log several types of information that the system generates during normal execution, such as data changes and actions/commands invoked by the end-users.

Structuring the UI web application to support a security token service (the Authentication, Authorization and Audit Logging component) leads to the architecture and protocols shown in Figure 44.

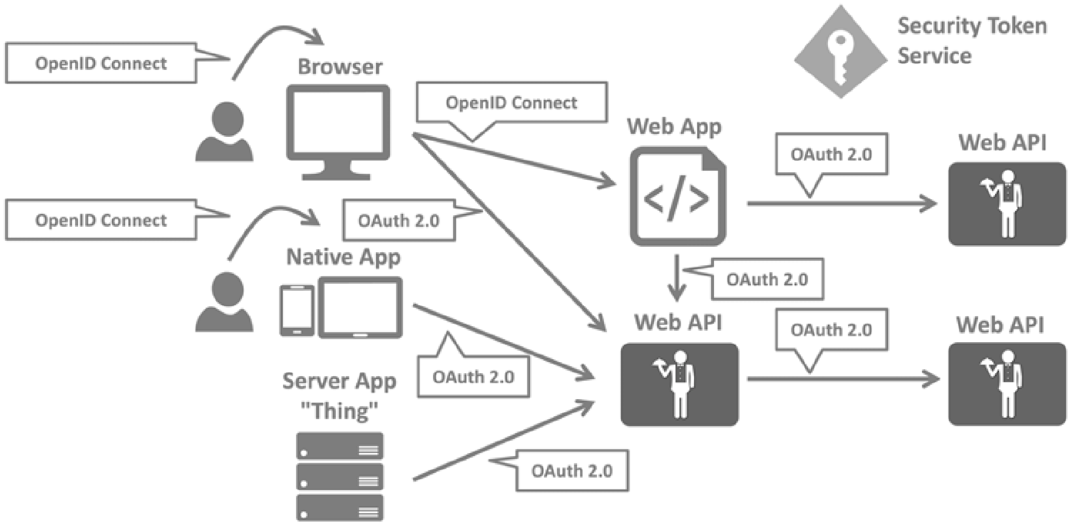

Figure 44: Security Token Architecture and Protocols.

The integration of OAuth-compliant REST API authentication and authorization between the Dashboard UI client and the Backend Coordinator b service represents a robust and secure framework that has been successfully implemented in our system. This implementation enhances security by ensuring secure authentication, fine-grained authorization, token-based security, secure communication, client credentials, token revocation, and adherence to standardization and best practices.

Secure authentication is achieved with OAuth, which replaces the direct sharing of sensitive credentials like usernames and passwords with the issuance of access tokens. These tokens serve as proof of authentication and are included in subsequent API requests to the Backend Coordinator backend service, ensuring secure access to protected resources. The Identity Server issues access tokens during the OAuth authentication process, providing a secure and efficient way to authenticate users.

Fine-grained authorization is another key aspect of our OAuth implementation. It empowers users to grant specific permissions (scopes) to the Dashboard UI client, dictating the actions that the client can perform on behalf of the user. This ensures that only authorized operations are executed, as the Dashboard UI client requests access to specific scopes during the OAuth authentication process, which are validated by the authorization server (Identity Server). The resulting access tokens contain the appropriate permissions, limiting the client's access to only authorized resources and functionalities within the Backend Coordinator backend service.

Token-based security is a crucial aspect of our system, as it relies on access tokens for secure communication between the Dashboard UI client and the Backend Coordinator backend service. These tokens are short-lived and cryptographically signed, minimizing the risk of unauthorized access and data breaches. Access tokens serve as temporary credentials, granting access to protected resources for a limited duration. Upon expiration, the Dashboard UI client obtains new tokens through the OAuth authentication process, reducing the window of vulnerability and enhancing security.

Secure communication is enforced with HTTPS (HTTP Secure), which encrypts data transmitted between the Dashboard UI client and the Backend Coordinator backend service. This encryption ensures the confidentiality and integrity of sensitive information, including access tokens and user data, mitigating the risk of eavesdropping and tampering. Our OAuth implementation also supports client credentials, allowing the Dashboard UI client to authenticate itself directly with the authorization server (Identity Server). This mechanism verifies the identity of the client, ensuring that only registered and trusted clients can access protected resources in the Backend Coordinator backend service.

Token revocation is another important feature of our OAuth implementation. It enables users to invalidate access tokens if unauthorized access or token compromise is suspected, providing a proactive approach to security that empowers users to mitigate risks associated with unauthorized access and data breaches. Finally, our OAuth implementation adheres to industry-standard protocols and best practices for authentication and authorization, ensuring interoperability, consistency, and adherence to industry security guidelines.

In summary, our OAuth-compliant REST API authentication and authorization implementation, utilized between the Dashboard UI client and the Backend Coordinator backend service, exemplifies a robust and secure framework that prioritizes user privacy, data security, and regulatory compliance.

4.2.4 Two-Factor Authenticator of Dashboard UI

To enhance the security access of the Dashboard UI, a Two-Factor Authentication feature using Google Authenticator has been integrated with the existing Identity Server OAuth service. Two-factor authentication (2FA) on web applications works as follows: The user first logs in with their username and password, which represents the first authentication factor - something they know. After successfully entering the username and password, the web application then prompts the user to provide a second form of authentication, such as a one-time code sent to their registered mobile device. This one-time code represents the second authentication factor - something the user has.

The user receives the one-time code, typically via SMS or a mobile app, and enters it into the web application to complete the login process. Once the user provides the correct one-time code, they are granted access to the web application.

The key aspect of how 2FA works is that it requires two independent factors to authenticate the user - something they know (password) and something they have (mobile device). This provides an extra layer of security beyond just a username and password, making it much harder for an attacker to gain unauthorized access. Common 2FA methods include one-time codes sent via SMS, mobile app authenticators, and hardware security keys. Implementing 2FA is recommended by security experts to better protect user accounts and sensitive data on web applications.

(1) Sign in (2) Step Verification

Figure 45: Two-Factor Authentication feature.

Google Authenticator stands out as a preferred option for two-factor authentication (2FA) due to several key advantages. Its utilization of a time-based one-time password (TOTP) algorithm enhances security by generating unique, time-sensitive codes that are challenging for attackers to predict or intercept. This dynamic approach surpasses static codes or SMS-based methods, bolstering the overall security posture.

Moreover, the ease of use associated with Google Authenticator contributes to its popularity. Available as a free app on both Android and iOS platforms, its user-friendly interface and straightforward setup process streamline the user experience. By enabling users to effortlessly set up accounts through QR code scanning or manual key entry, Google Authenticator minimizes complexity and hardware requirements, fostering accessibility.

Additionally, Google Authenticator's offline functionality sets it apart from other 2FA solutions. Operating independently of internet connectivity, the app ensures reliability in scenarios with limited or no network access. This autonomy enhances user convenience and security, making it a dependable choice for safeguarding accounts across various platforms.

Furthermore, the widespread compatibility of Google Authenticator with numerous online services underscores its versatility. Embraced by major platforms like Google, Facebook, and Dropbox, its broad adoption enhances its applicability and convenience for users seeking consistent 2FA protection across diverse services.

The credibility of Google as a reputable company further reinforces the trustworthiness of Google Authenticator. Known for its commitment to security and privacy, Google's regular updates to address vulnerabilities and enhance functionality instill confidence in users, solidifying Google Authenticator as a reliable 2FA solution.

While Google Authenticator offers compelling benefits, it's essential to acknowledge the existence of alternative 2FA options like Authy, Microsoft Authenticator, or hardware tokens such as YubiKey. The selection of a 2FA method may hinge on factors like user preference, system compatibility, and specific security needs. Ultimately, the primary objective of 2FA remains to fortify security by mandating dual authentication, with Google Authenticator emerging as a popular and effective choice for achieving this objective.

4.2.5 WebSocket of Dashboard UI

In our system, real-time data presentation between the Backend Coordinator and the Dashboard UI is facilitated through SignalR, a library that implements the WebSocket protocol. This implementation prioritizes security to safeguard sensitive information exchanged in real-time. The following details highlight the security aspects of this setup:

Encrypted Communication: SignalR WebSocket connections are established over HTTPS, ensuring encrypted communication between the Backend Coordinator and the "Dashboard UI." This encryption mechanism utilizes SSL/TLS protocols to protect data from interception and tampering, maintaining the confidentiality and security of real-time information.

Same-Origin Policy (SOP): SignalR WebSocket connections adhere to the same-origin policy, restricting communication between scripts from different origins. By enforcing SOP, SignalR mitigates the risk of cross-site scripting (XSS) attacks, ensuring that real-time data remains isolated and secure within the application's origin.

Built-in Support for Secure Protocols: SignalR WebSocket communication supports secure protocols like TLS, enhancing data encryption, authentication, and integrity protection. Leveraging TLS ensures that WebSocket connections are secure, resistant to attacks, and safeguarded against eavesdropping and data tampering.

Authentication and Authorization: SignalR integrates with various authentication mechanisms to verify client identities and enforce access control policies. This enables the Backend Coordinator to authenticate users, control data access based on permissions, and implement secure authentication mechanisms tailored to the application's needs.

Server-Side Security Measures: The SignalR WebSocket server implements additional security measures such as input validation, rate limiting, and secure coding practices to protect against attacks and vulnerabilities. Proactive security measures at the server level ensure the integrity and robustness of real-time data transmission.

Cross-Origin Resource Sharing (CORS): SignalR WebSocket communication supports CORS, allowing the Backend Coordinator to specify trusted origins for accessing real-time data. By defining CORS policies, SignalR enhances security, prevents unauthorized cross-origin requests, and ensures WebSocket connections are established only from trusted sources.

Secure Deployment Configurations: Proper configuration and securing of the SignalR WebSocket server infrastructure are crucial for ensuring the overall security of real-time data transmission. Implementing firewall rules, network segmentation, and intrusion detection systems, along with regular security audits and updates, help maintain the integrity and security of the WebSocket server environment.

In summary, SignalR WebSocket communication between the Backend Coordinator and the Dashboard UI incorporates multiple layers of security measures to ensure the confidentiality, integrity, and availability of real-time data. By leveraging encryption, authentication, access control, and other security features, SignalR enables the development of robust and secure real-time web applications that adhere to the highest security standards and compliance requirements.

4.3 MQTT Integration

Within the Dashboard UI, an MQTT architecture is utilized with a central MQTT broker facilitating a publish-subscribe mechanism for data ingestion from three distinct sources: UAV platform, satellite, and the legacy systems.

Publisher-Subscriber Interaction

- **Publishers**: UAV platform, satellite component, and HDE LSI act as publishers, generating messages with data for sharing.
- **Subscriber**: Backend Coordinator of the dashboard system is the sole subscriber, issuing subscription requests to the MQTT broker to receive data.

4.3.1 Integration with satellite component

The integration involves the Satellite Component, which is part of the Backend Coordinator system, interacting with the Dashboard UI to facilitate the prediction process based on user-defined areas of interest. We concluded with the integration as shown in Figure 46.

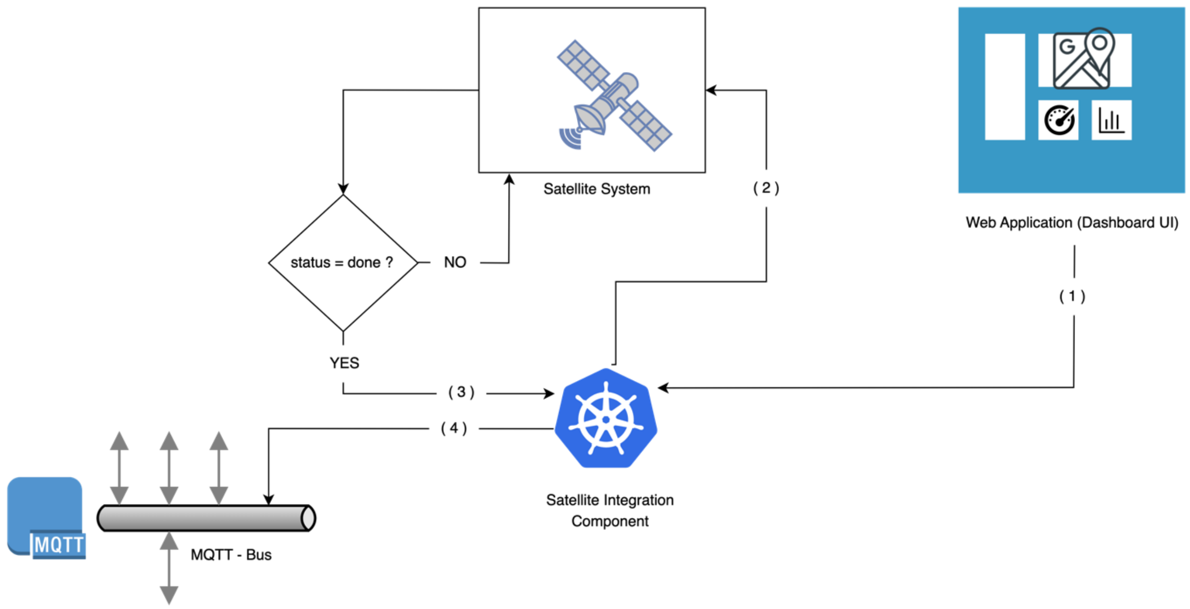

Figure 46: Satellite Component Diagram.

1. **User Interaction and Data Transmission**:
    - The end user utilizes the UI Dashboard to define areas of interest, marking them as Point or Polygon objects.
    - This information is transmitted to the Satellite Component of the Backend Coordinator for further processing.
2. **Prediction Request Initiation**:
    - The Satellite component triggers a "prediction request" by sending a POST request to the /prediction endpoint.

- The request body includes a GEOJSON FeatureCollection with Point and Polygon objects for prediction.
- Upon receiving a response, the job_id is stored for future reference in retrieving results.

3. **Status Updates and Result Retrieval**:
   - The Satellite component uses the stored job_id to fetch status updates and results by sending a GET request to the /prediction/{job_id} endpoint.
   - The response includes a JSON object with fields like status (current state of the prediction job), history (progress tracking), and messages (results related to objects in the prediction request).
4. **Progress Monitoring**:
   - Continuously monitors the prediction job progress by querying the /prediction/{job_id} endpoint until the job is completed (status = done).
5. **Completion and MQTT Message Publication**:
   - Upon job completion, retrieves the "messages" section of the latest results.
   - Based on this data, the Satellite component generates and publishes an MQTT message to the MQTT broker for further processing.

This seamless integration process ensures that data from the Satellite inspection tool is effectively processed, monitored, and shared through the MQTT system for efficient communication and data dissemination within the system.

4.3.2 Integration with UAV component

The integration involves the UAV Component, a recent addition to the Dashboard's internal infrastructure, collaborating with the UAV Platform and UAV Detection System to enhance the detection process. We concluded with the integration as shown in Figure 47.

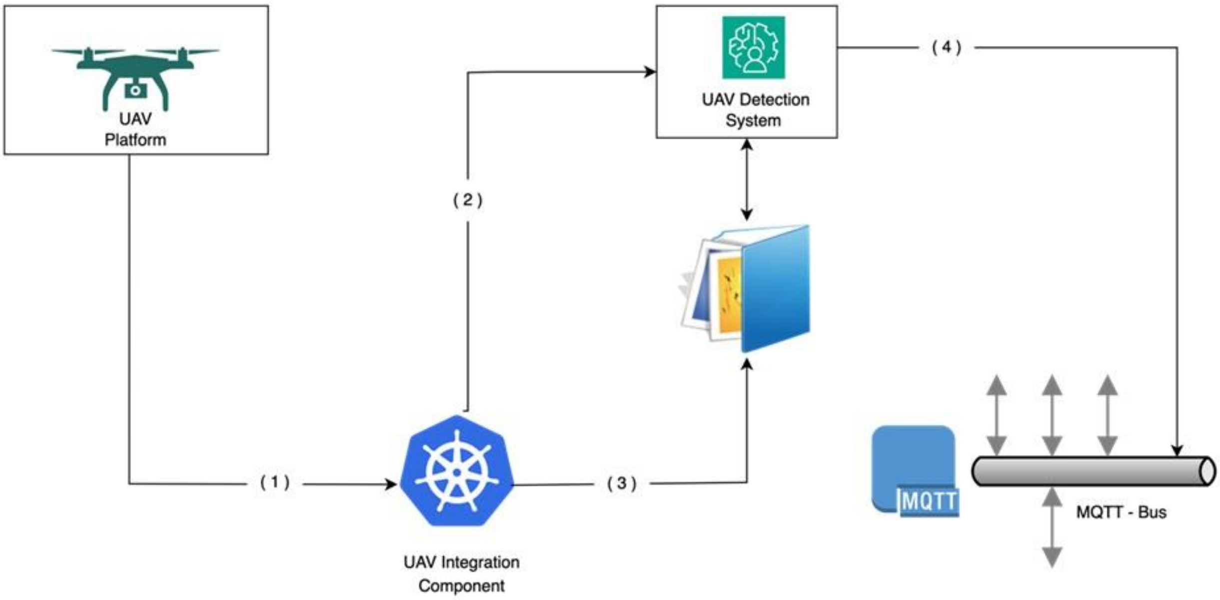

Figure 47: UAV Integration Component Diagram.

1. **Interaction with the UAV Platform:**
    - The UAV component interfaces with the UAV Platform to obtain both live video/images and stored video/images.
2. **Interaction with the UAV Detection System:**
    - The component initiates the UAV Detection System through a REST API POST call, providing settings for the UAV visual inspection routine and source details to enable real-time inspection.
    - Leveraging stored videos/images from the UAV Platform, the UAV Component sends requests and organizes files in an input folder.
    - The UAV Detection System scans all videos and images within the input folder, searching for "positive" results from detection pipelines.
    - Upon detection, the UAV Detection System generates and publishes messages containing the results of the detection algorithms on the MQTT broker.

4.3.3 Legacy systems integration

Hydro Dolomiti Energia (HDE) is the sole partner who has engaged in discussions and file exchanges to enable the running of the UI Dashboard on their legacy systems. We concluded with the integration as shown in Figure 48.

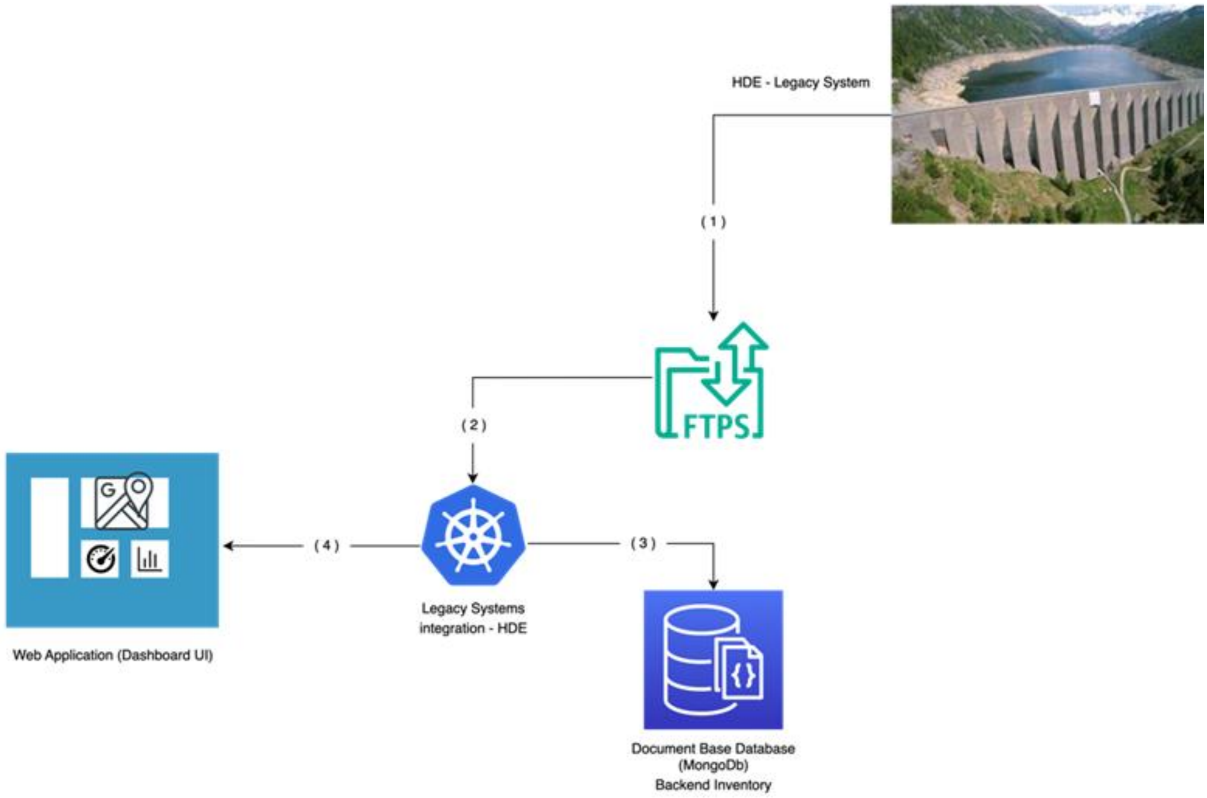

Figure 48: HDE - Legacy System Integration Diagram.

1. The HDE Legacy System, as part of its operations, creates a JSON file containing relevant data and then proceeds to transfer this file to the FTPS (File Transfer Protocol Secure) server for storage and further processing.
2. Within the system architecture, the Legacy Systems Integration (LSI) component plays a crucial role by actively monitoring a specific folder on the FTPS server. This monitoring function is designed to keep track of any new files that are deposited into this designated folder.
3. When the LSI component detects the presence of a new file in the monitored folder, it initiates a process to archive this file within the Backend Inventory system. This archival step is essential

for maintaining a comprehensive historical record of data, enabling in-depth analysis and insights over time.

4. In addition to archiving the incoming files, the LSI component is programmed to extract specific data elements known as "Properties of Interest" from the received file. These extracted properties are then transmitted to the dashboard User Interface (UI) for visualization purposes. The UI leverages this data to dynamically display the relevant information on a map, associating each property of interest with its corresponding marker for easy interpretation and analysis.

4.3.4 Legacy systems integration security

Our system implementation has been integrated with the existing legacy systems, fortified with the secure File Transfer Protocol over SSL/TLS (FTPS). Below are some reasons why FTPS is considered secure:

Encryption: FTPS utilizes encryption to safeguard data in transit. It employs SSL/TLS (Secure Sockets Layer/Transport Layer Security) protocols to encrypt the connection between the Legacy Systems' client and server. This encryption ensures that data exchanged between the client and server cannot be intercepted or tampered with by unauthorized parties.

Authentication: FTPS supports various authentication methods to verify the identities of both the client (Legacy Systems) and server. These methods include username/password authentication, client certificates and server certificates. By requiring authentication, FTPS ensures that only authorized users and servers can access the data being transferred.

Data Integrity: FTPS verifies the integrity of data during transmission to ensure that it has not been altered or corrupted. This is achieved with cryptographic hash functions, which generate checksums or hash values for each data packet. The recipient can verify the integrity of the data by comparing the received hash value with the expected value.

Server Authentication: FTPS servers are typically required to present a digital certificate issued by a trusted Certificate Authority (CA). This certificate contains information about the server's identity and is used to authenticate the server to the Legacy Systems client. By verifying the server's certificate, the client can ensure that it is connecting to the correct server and not a malicious imposter.

Client Authentication: In addition to server authentication, FTPS can also require client authentication using digital certificates. This provides an additional layer of security by verifying the identity of the Legacy Systems client before allowing access to the server.

Firewall Friendly: FTPS is designed to work seamlessly with firewalls and network address translation (NAT) devices. It uses a single port (typically port 21 for control connections) for communication, making it easier to configure and manage firewall rules.

Compliance: FTPS implementations often adhere to regulatory compliance standards such as PCI DSS (Payment Card Industry Data Security Standard) and HIPAA (Health Insurance Portability and Accountability Act). Compliance with these standards ensures that sensitive data is protected during transmission, helping organizations like Legacy Systems meet their legal and regulatory requirements.

Overall, FTPS provides a secure and reliable method for Legacy Systems to transfer files over a network, making it suitable for use in environments where data security is a priority. By employing encryption, authentication, data integrity checks and compliance with industry standards, FTPS helps Legacy Systems protect their sensitive data from unauthorized access and interception.

4.4 Deployment

The Eclipse Mosquitto MQTT broker and the MongoDB database are key components of the overall system architecture, providing the necessary messaging and data storage capabilities to support the various subsystems and the Dashboard UI.

MQTT Bus Service: This involves installing the Eclipse Mosquitto MQTT broker, version 5 or 3.1.1, on the cloud platform, configured to listen on default ports (1883). The initial setup of the broker includes creating two main topics: "UAV" for use by the UAV detection subsystem and "SAT" for use by the Satellite Component. Both subsystems publish the results of their detection processes to these topics.

Document-Based Database (MongoDB) for Backend Inventory: This entails installing the latest version of MongoDB on the cloud platform, configured to listen on default ports (27017). The initial setup of the database includes a database named "Sunrise" with two collections: "UAV" for storing all detection events from the UAV detection subsystem and "SAT" for storing all detection events from the Satellite Component. These events are later utilized for reporting and historical data visualization on the Dashboard UI.

4.4.1 Security on MQTT protocol

In our design, a comprehensive suite of security features has been integrated to enhance the integrity, confidentiality, and reliability of our MQTT communication protocol. These measures collectively fortify the security posture of our system, ensuring data protection and secure communication channels.

Transport Layer Security (TLS): The deployment of Transport Layer Security (TLS) encrypts our MQTT communication channels, safeguarding data from interception and tampering. TLS certificates, diligently managed and updated, authenticate clients and brokers, ensuring connection authenticity and enhancing defense against security breaches.

Authentication: Our infrastructure enforces strict client authentication protocols through methods like username/password authentication, client certificates, and OAuth tokens. These mechanisms allow only authorized clients with valid credentials to establish connections, reducing the risk of unauthorized access to sensitive data.

Access Control Lists (ACLs): Carefully configured Access Control Lists (ACLs) establish granular access control policies, governing access to specific MQTT topics based on predefined permissions. By implementing ACLs, access to sensitive data is restricted, preventing unauthorized clients from accessing beyond their designated scope.

Message Encryption: Clients utilize robust application-level encryption techniques to encrypt message payloads before publishing to the MQTT broker. This additional encryption layer ensures message content confidentiality, protecting sensitive information from unauthorized access.

Broker Configuration: Our MQTT broker is configured to enforce stringent security policies and mitigate common threats. Regular security audits and updates are conducted to ensure resilience against emerging vulnerabilities and threats.

Secure MQTT Implementations:

Leveraging secure MQTT broker implementations known for robust security features and proactive vulnerability management, our client libraries and SDKs are equipped with comprehensive security measures. This empowers developers to build secure MQTT applications confidently.

By implementing and maintaining these security measures, a robust and resilient MQTT infrastructure has been established, prioritizing the security and integrity of data and communication channels.

5 Pilot trials execution

5.1 Elektro-Slovenija, d.o.o. (ELS)

5.1.1 Description of the pilot

Pilot 1 was conducted on the 110kV Ribnica-Kočevje power line within the VLOS scenario. To implement Pilot 2 in the BVLOS scenario, the location is contingent upon following requirements:

- Each UAV operator must be registered in accordance with legislation of the Slovenian Civil aviation Authority. Since the drone is equipped with a camera and has a weight exceeding 250 grams, successful registration within the Slovene CAA includes a qualified digital certificate and requires the payment of a fee.
- Operators must complete appropriate training and pass an exam, depending on the category and subcategory of the drone. For specific operations such as BVLOS**,** additional training and certification are required in accordance with Slovene CAA.
- Conducting BVLOS flights requires obtaining an operational permit from the CAA. This includes submitting a detailed operational plan, a risk assessment and proof of the operator's qualifications.
- Operators must adhere to airspace restrictions, such as no-fly zones near airports, overpopulated areas and protected areas like Triglav National Park. Special permits are required for flying in such areas.
- It is recommended that operators obtain insurance to cover potential damages to third parties that may occur during drone operations.
- As the operators are foreign, it will also be necessary, as before, to arrange a “cross-border operation” with the Slovenian CAA.

By completing all the above-mentioned steps, Eles can officially select the pilot site, preferably somewhere in the rural area (similarly to Pilot 1).

Approximately 59% of Slovenia is covered in forests, meaning that powerlines tend to cross various regions, ranging from flat fields and meadows all the way to hilltops. Field and meadows do not present an issue, but when it comes to inspecting powerlines where the terrain is steeper, a drone flight would solidify and enhance the procedure of inspection of a certain issue or detect any potential anomalies within the regularly scheduled checkup, especially in terms of enhanced access to the point of interest.

5.1.2 Description of End-Users' Roles

End users in Eles’s case will represent the team that has acquired all the licenses needed to fly within the specified area (including controlled geographical areas) with respect to the all the requirements needed to officially announce an operation. All the designated pilots will also have full authority and clearance to analyze the data required from the piloting activities.

Note that some of the end users mentioned in the table below may or may not correspond to the same function (user role).

Table 6: ELS - roles and profiles used when using the tool.

User organization	End user profile	User Role	Skills
ELES	*Strategic*	*Chief Technical Officer*	*Complete and comprehensive knowledge of the sector (policies, inspection requirements, technologies)*
ELES	*Tactical*	*Head of Operative Unit*	*Technical (drone operator), data analyst*
ELES	*Tactical/ Operational*	*Technical system specialist*	*Technical (drone operator), data analyst*
ELES	*Operational*	*Data analyst*	*Configurator and interpreter of input data, expert knowledge on tech, drone operator*

5.1.3 CI's evaluation of the SUNRISE RII Tool

These tools will offer Eles various and variable insights regarding the inspection of its infrastructure.

The Satellite inspection tool has shown the promising results of the data captured. Similar to the UAV, these satellite images can detect possible anomalies much faster, such as illegal buildups within the vicinity of the power line. Accessing this tool, which efficiently captures anomalies in the vicinity of the power infrastructure, enables the appropriate technical (maintenance and similar) teams to react accordingly to the situation at hand. Likewise, the way that the UAV tool uses all the available data captured by the drone, combined with the software behind it, offers a much more time efficient procedure in analyzing the power infrastructure.

5.1.4 Financial needs for adoption of the tool

The financial investments are contingent upon the number of licensed pilots that would be trained to be considered fully operational drone operators, which include (from an individual perspective) the purchase of a specific drone (equipped with specific cameras, sensors and other attachments), proper licensing to have the capability to fly within Eles's infrastructure, and the adaptability to include any supporting hardware (and software) into the pre-existing or currently operational systems.

Since Pilot 2's UAV is expected to be within the BVLOS scenario, the cost for the initial set up, assuming the pilot is performing the BVLOS flight plan from ground zero, remains the same. However, all estimated financial costs are estimations. The starting adoption of license(s) remains the same in UAV preparation, which involves up to 500 hours of training, and the cost of purchasing drones (and the supporting equipment) is up to approximately 50,000 EUR. The UAV preparation is the same; it will take up to 60 hours for any specific scenario, including calibrating the equipment and all the preemptive maintenance and repairs. Where the numbers differ is within the training of the licensed pilots in the BVLOS scenario. Pilot 1 took place in the VLOS scenario, which meant that the pilot had all the requirements. The correct license to pilot the drone was within the pre-acquired authorization by Eles. On the other hand, training for such a specific scenario would require specific authorizations such as the ones mentioned in chapter 5.1.1. While the VLOS scenario includes a potential training time of approximately 45 hours, BVLOS, especially with the licenses and requirements that need to be achieved, is estimated to take approximately 100 hours.

Satellite imagery is dependent on the providers of such images, meaning that it varies based on the quality of the data (images) and the price for the specific solution. Similar to UAV, assuming that the implementation for such images is ground zero, such images can cost up to 20,000 EUR/km, and although we might assume that these images are perfect, there are still various factors that do not fully utilize the investment, such as bad weather and a consequently reduced database of images that an infrastructure operator may use. Regardless, the cost estimation remains the same as it was calculated in Pilot 1.

5.1.5 What are the benefits of the remote inspection tool?

The usefulness of this tool is considered within the context of inspecting the power lines and other energy infrastructure.

Rural areas where the power lines go through might be in a relatively remote location, making access to them challenging. By implementing drones, the required (trained) personnel would then announce and perform the flight paths along the power line, investigating (preemptively or intentionally/planned) any potential anomalies on the pylon, isolators, cables and similar key points.

However, while the time it takes to use the drones and satellite imagery for the inspection of the infrastructure is evidently reduced, when detecting a potential anomaly, designated teams would still have to announce the maintenance procedures and predict the (temporary) shutting down of a specific section on the power lines or the substations, and then dispatch the maintenance personnel to the site of the anomaly.

5.2 Slovenske Železnice (SZ)

5.2.1 Description of the pilot

The location of the SZ was between the railway stations Gornje Ležeče (km 657+503) and Divača (km 669+565) and located on the main railway line no. 50 Ljubljana- Sežana d.m. The line is double-track and electrified. The longitudinal profile is in the direction of Gornje Ležeče – Divača with a gradient of 2.46‰ to 6.94‰. Distance from Ljubljana is approximately 70 km.

The second location was on station Rakek (in km 621+211), which is also located on the main railway line no. 50 Ljubljana- Sežana d.m. Distance from LJ is approximately 50 km.

The first location was selected because the terrain along the track is difficult to access, fire incidents occur along the railway track and there is a long response time when such events occur.

The second location was selected because the railway station Rakek has a high density of tracks and a complex catenary, and it was possible to inspect the whole system from one point of view.

In both locations, it was possible to perform inspection with VLOS UAV flights.

The Ministry of infrastructure of Slovenia (MZI) will continue to help obtain authorization for the unmanned aerial vehicle (UAV) pilot trials that are planned in Slovenia in the summer of 2025. The Ministry offers itself as the link between the Civil aviation agency of Slovenia and Slovene CI operators (SZ, ELES), as well as SKYLD who will conduct the pilot flight trials for the visual infrastructure inspection with UAVs.

5.2.2 Description of End-Users' Roles

End users in SZ's case represented a team that has only operational knowledge about railway traffic and specific maintenance for catenary. Currently, there are no UAV certified operators in SZ. The roles are presented in the table below.

Table 7: SZ - roles and profiles used when using the tool.

User organization	*End user profile*	*User Role*	*Skills*
SZ	*Strategic*	*Technical Expert*	*Complete and comprehensive knowledge of the sector (policies, inspection requirements, technologies)*
SZ	*Tactical*	*Head of Operative Unit*	*Technical expert for maintenance of the catenary*
SZ	*Operational*	*Train management dispatcher*	*Knowledge on TMS*

5.2.3 CI's evaluation of the SUNRISE RII Tool

The UAV-based inspection tool within SUNRISE has demonstrated significant potential in enhancing infrastructure monitoring. The pilot demonstration showed that high-definition cameras can successfully detect infrastructure anomalies as well as possible fires and landslides. Image visualization was sufficient and precise enough to detect fire, cracks on catenary and especially on isolators, due to the powerful multi-zoom camera.

MZI is a government regulatory body that supports the adoption of technology or tools that would benefit the operator of the critical infrastructure.

5.2.4 Financial needs for adoption of the tool

Currently, no estimates were made.

5.2.5 What are the benefits of the remote inspection tool?

In recent years, the use of drones for remote infrastructure inspection has revolutionized how we monitor and maintain critical infrastructure. Drones equipped with high-resolution cameras, thermal sensors and LiDAR technology offer numerous advantages over traditional inspection methods, which often require costly and hazardous manual labor. Checkups of infrastructure defects, anomalies or hazards require additional manpower and time, which in potential future pandemics would bring additional difficulties due to the lack of manpower.

Inspection must consider the ongoing traffic. Also, access to different sections of the railway is often difficult. Drone inspection of the railway line (and overhead catenary) can be carried out regardless of the ongoing traffic, thus increasing the safety of the inspection. Regular drone inspections can cover large areas, save time and allow for faster detections of defects and hazards such as landslides, fire, foreign objects on the railway, catenary defects etc., thus preventing accidents and reducing congestion and delays.

5.3 Hydro Dolomiti Energia, s.r.l. (HDE)

5.3.1 Description of the pilot

The CIs selected for the Pilot 2 are the six hydropower intake weirs of the Leno Valley. This is a side valley of Daone Valley, in Trentino Region, Italy. These intake weirs divert the water to the Boazzo hydropower plant, one of the major power plants of the area, providing reliable and renewable energy production. The choice of these CIs was mainly driven by their accessibility: in fact, their location makes them very difficult to access, reducing the inspection frequency and the awareness regarding their status. The same reduction of manpower availability on-site happens on all the other CIs during pandemics; thus, both conditions could benefit from the same technological solution.

This final Pilot will demonstrate the real potential of a Remote Inspection Tool, composed by an Unmanned Aerial System and a AI-based visual inspection tool. In particular, will demonstrate the capability of a UAS to fly autonomously in a remote and mountainous area and perform a visual inspection of a CI, and of an AI-based to effectively process the content of the images captured by the UAV payload, providing accurate and reliable information.

The mountainous area, which is the typical operational environment of HDE, poses serious challenges to any asset, including aerial vehicles: rapidly-changing weather and more extreme events with respect to plains; low temperatures; orographic obstacles to visual and radio line of sight; reduced coverage of mobile phone signal. Last but not least, increasing the altitude, the reduced air density impacts the ability of aerial vehicles to generate adequate lift and reduce flight autonomy.

So, this final Pilot is pivotal in demonstrating that the developed technologies fulfill, in a real operational environment, the functional and non-functional requirements identified in the Tool's design.

5.3.2 Description of End-Users' Roles

Table below summarizes the personnel (their profile, role and required skills) that will be using the RII Tool in HDE. It is assumed that the Tool has reached maturity and has been deployed in the Company; all the required hardware (UAS, server for images processing with AI, …) is owned by HDE, and all the software (AI models, dashboard, …) has been licensed and is run by HDE in its IT environment.

Table 8: HDE - roles and profiles used when using the tool.

User organization	End user profile	User Role	Skills
HDE	*Strategic*	*Chief Executive Officer/Chief Technical Officer*	*Acts as UAS Operator, in accordance with relevant UAS regulation and in compliance of all its requirements. Has a complete and comprehensive knowledge of the sector (policies, inspection requirements, technologies)*
HDE	*Tactical*	*Head of Operative Unit*	*Technical (knows the features and the limits of the Tool), data analyst (is able to interpret the data gathered by the tool and the results it provides). Takes maintenance and operational decisions also on the base of the Tool's results.*
HDE	*Tactical/ Operational*	*Technical system specialist*	*Technical (knows the features and the limits of the Tool, can configure it and tune its parameters), data analyst (is able to interpret the data gathered by the tool and the results it provides, has the base knowledge needed to understand the AI models at the core of the Tool and to properly set their parameters for an optimized inspection), has a drone pilot license, can perform a UAS mission planning.*
HDE	*Operational*	*Operative personnel*	*Know-how of the maintenance and inspection tools and procedures. Receive the inspection results from the Tool and guarantee proper maintenance of the CIs. Subordinate to the Head of Operative Unit, takes maintenance and operational decisions also on the base of the Tool's results.*
HDE	*Operational*	*Technical support staff*	*Specialized technical knowledge in topography, structural engineering and other fields interested by inspections. Takes maintenance and operational decisions also on the base of the Tool's results. Receive the inspection results from the Tool and guarantee proper maintenance of the CIs. Takes maintenance decisions also on the base of the Tool's results.*
HDE	*Operational*	*IT staff*	*Specialized technical knowledge in IT and computer systems. Responsible of the installation and maintenance of the software modules of the RII Tool.*

5.3.3 CI's evaluation of the SUNRISE RII Tool

The RII Tool will be evaluated according to its ability to meet the already-defined functional and non-functional requirements in the real operational environment of HDE, as described in Chapter 5.3.1 "Description of the pilot". In fact, up to now, it was only possible to evaluate the RII Tool components in a controlled environment. This final test should demonstrate the integration of the components and their performance under the operational and environmental challenges posed by the real-world operations.

5.3.4 Financial needs for adoption of the tool

This is work in progress and dependent on the execution of Pilot 2. Namely, no costs of any component of the RII Tool have been shared with HDE yet.

Nevertheless, it's possible to identify the main cost components for the adoption and operation of the RII Tool: Capital Cost and O&M Cost. The higher is expected to be the Capital Cost: buy of UAS hardware (UAV with payload, batteries and all the required accessories; spare parts; ground control station), buy of hardware for execution of RII software; training of HDE's personnel for pilots and for using the Tool; cost of obtaining the proper flight licenses for the specific category (according to current regulations). Thus, the Capital Cost also depends on the number of personnel to train.

The O&M Cost is mainly the cost of the consumable and spare parts for the UAS (rotor blades, rechargeable batteries, etc.), the electricity for running the software and for recharging the batteries (almost negligible), the IT cost for running the RII software modules (server maintenance and repair, etc.).

On the other hand, a significant saving can be achieved with this Tool; it should be considered when evaluating the financial aspect of its adoption. In fact, the inspection of the selected CIs is currently performed by teams of two people, which are transferred to the site by helicopter. To complete the inspection of all the Leno Valley's CIs, at least 60 man-hours are needed, plus the cost of the helicopter (roughly 1000 € in total, per each complete inspection).

As of now, the only known cost is the training for obtaining pilot licensing, which is performed with third-party officially licensed instructors and depends on the type of mission. It can be evaluated in roughly 2k€/pilot.

5.3.5 What are the benefits of the remote inspection tool?

The benefits are:

- Higher frequency of inspection.
- Lower cost of inspection.
- Less workload for human operators involved in inspections/ lower dependency on them in case of reduced crew availability (e.g. pandemics).
- Human operators are less subject to the risks associated with manned inspections in rough and isolated areas.
- Quicker updates on the status of the CIs, e.g. after weather events.
- Better maintenance and operative planning, thanks to better information.

5.4 ACOSOL (ACO)

5.4.1 Description of the pilot

The ACOSOL pilot is conducted in water treatment, distribution facilities and supply networks in Spain.

Water is a critical infrastructure essential for public well-being, especially during crisis situations such as pandemics. This area was selected because ensuring a continuous water supply is vital for public health, hygiene, and the maintenance of essential services. Additionally, traditional inspections in these facilities can be complex and costly, making the implementation of remote inspection tools highly beneficial.

ACOSOL expects the pilot to provide:

1. **Optimized inspection and maintenance:** Use of drones and remote sensors to monitor infrastructure conditions without requiring constant physical inspections.
2. **Reduced response times:** Early detection of issues in pipelines, pumping stations, deposits and treatment plants.
3. **Increased operational efficiency:** Lower costs associated with manual inspections and improved preventive maintenance.
4. **Enhanced service resilience:** Ensuring continuous water supply during emergencies without compromising staff safety.

Ensuring the uninterrupted operation of the water distribution network is essential for social stability and public health. During a pandemic or any disruptive event, reducing the physical presence of personnel in facilities without compromising service quality becomes a key factor. Implementing advanced remote inspection tools improves incident response capabilities and ensures ACOSOL's operational sustainability.

5.4.2 Description of End-Users' Roles

In the real execution of the ACOSOL pilot, active participation will primarily involve:

- **José María Jiménez (ACOSOL employee)** – Key ACOSOL representative overseeing the implementation and evaluation of the remote inspection tool.
- **Local Police of Marbella (drone support officer)** – Provides aerial support with UAV operations for infrastructure inspection.
- **Additional operational support staff** – Assists in the coordination and execution of the pilot, ensuring the necessary conditions for testing the remote inspection tool.

While additional personnel have been involved in planning and consultation phases, their presence during the execution will be limited. These roles include:

- **Water Treatment and Distribution Managers** – Participated as needed in preparatory stages to ensure alignment with ACOSOL's operational goals.
- **IT and Cybersecurity Specialists** – Engaged in an advisory capacity when required to assess data security and integration aspects.
- **Operations Managers** – Contributed insights during the planning phase but will not be actively present during the on-site testing.

This distinction between real and ideal participation highlights the practical execution of the pilot while acknowledging the broader team's involvement in shaping the project. Despite the limited on-site presence, the pilot benefits from prior strategic input from key stakeholders. The table below reflects the end-user profile for ACO:

Table 9: ACO - roles and profiles used when using the tool.

User organization	End user profile	User Role	Skills
ACO	*Strategic Chief Technical Officer*	*Supervises the integration of the tool into ACOSOL's strategy.*	*Knowledge of infrastructure policies, UAV inspection technologies, and AI-driven analytics.*
ACO	*UAV Drone Operator (Local Police of Marbella)*	*Conducts aerial inspections using high-resolution, thermal, and LiDAR-equipped drones.*	*UAV pilot certification, experience with BVLOS operations, and knowledge of AI-based anomaly detection.*
ACO	*Data Analyst*	*Analyzes data to detect leaks, blockages, and vegetation growth.*	*AI-based data interpretation, infrastructure analytics, and software proficiency.*
ACO	*Water Treatment & Distribution Manager*	*Ensures integration of inspection results into maintenance workflows.*	*Technical understanding of water treatment processes and predictive maintenance.*
ACO	*Emergency Response Coordinator*	*Uses inspection insights to develop response strategies for infrastructure failures.*	*Risk management, crisis response planning, and data-driven decision-making.*

5.4.3 CI's evaluation of SUNRISE RII Tool

ACOSOL considers the SUNRISE Remote Infrastructure Inspection (RII) Tool a valuable innovation for improving the monitoring and maintenance of its water infrastructure. The tool enhances inspection processes by providing remote access to real-time data, reducing the need for manual site visits, and allowing for a more proactive approach to infrastructure management.

Although ACOSOL has selected a limited number of sites for the pilot tests, a full-scale implementation presents considerable challenges:

- **Operational Complexity:** The vast majority of UAV flights required for a real-world deployment would need to be executed beyond visual line of sight (BVLOS). Given that the primary distribution network extends over 200 km, continuous monitoring would necessitate intermediate charging stations and signal relay points to maintain connectivity and operational efficiency.
- **Infrastructure Adaptation:** While feasible, deployment would require leveraging existing ACOSOL facilities for charging and communication relays, adding logistical and financial considerations.
- **Cost of Equipment and Training:** Full implementation would demand a significant investment in UAV equipment and specialized training for pilots to operate BVLOS missions safely and in compliance with aviation regulations.

While the technology is viable and aligns with ACOSOL's long-term strategic vision, its large-scale application would require substantial financial and operational planning to address these challenges.

ACOSOL has identified the following components of the tool as particularly beneficial:

- UAV-based remote inspection – Provides detailed and high-resolution images of infrastructure conditions, reducing the need for physical site visits.
- AI-powered anomaly detection – Enables early identification of potential issues such as leaks, structural weaknesses, or vegetation encroachment.
- Real-time data visualization – Enhances decision-making by providing immediate access to inspection results via an intuitive user interface.

Despite the challenges of large-scale implementation, ACOSOL sees the SUNRISE tool as a highly promising solution for improving the efficiency, resilience, and sustainability of its water infrastructure monitoring operations.

5.4.4 Financial needs for adoption of the tool

The adoption of the SUNRISE Remote Infrastructure Inspection (RII) Tool at ACOSOL would require financial investment across multiple areas, particularly in UAV technology and personnel training.

Hardware Costs:

- Primary UAV for Overflights:
  - Equipped with high-resolution optical cameras, thermal imaging, and AI-powered sensors for leak detection and structural assessment.
  - Recommended: LiDAR for vegetation growth monitoring near pipelines. However, vegetation analysis could alternatively be performed using high-resolution cameras and change detection software, reducing costs.
- **Secondary UAVs for Indoor Inspections:**
  - Small-sized UAVs with anti-collision sensors for the inspection of sewer and wastewater networks.
  - These drones provide an internal view of pipelines, detecting structural issues such as cracks, blockages, or corrosion in underground infrastructure.

Personnel Training Costs:

- **UAV Pilot Certification & Training:**
  - BVLOS (Beyond Visual Line of Sight) certification for personnel operating long-range UAVs.
  - Training in drone operation for indoor inspections, particularly in confined spaces like sewer systems.
- **Data Analysis Training:**
  - Staff will need training on AI-powered detection systems to interpret thermal imaging, high-resolution video, and LiDAR data.

Software & Licensing Costs:

- AI-driven change detection software for analyzing aerial and indoor inspection footage.
- Cybersecurity solutions for secure data transmission and integration into ACOSOL's existing monitoring platforms.

Despite the initial investment, ACOSOL anticipates several long-term financial benefits:

- **Reduced On-Site Inspections:** UAVs will significantly reduce the need for manual inspections, leading to lower labor and operational costs.
- **Optimized Preventive Maintenance:** AI-powered monitoring will enable early detection of leaks, corrosion, and pipeline obstructions, reducing emergency repair expenses.
- **Efficient Vegetation Management:** Remote detection of abnormal vegetation growth near pipelines will help prevent leaks, root intrusions and infrastructure damage.
- **Enhanced Safety & Workforce Efficiency:** Minimizing personnel exposure to hazardous environments, particularly in wastewater and confined space inspections, will lower risks and associated costs.

Overall, the combination of aerial and indoor UAV inspections presents a comprehensive and cost-effective solution for ACOSOL's infrastructure monitoring needs. While the implementation will require a considerable upfront investment, the expected savings in maintenance, labor, and emergency response justify its adoption.

5.4.5 What are the benefits of the remote inspection tool?

The adoption of the SUNRISE Remote Infrastructure Inspection (RII) Tool offers a wide range of benefits beyond cost reduction, particularly in enhancing operational efficiency, safety, and resilience.

Key Benefits Beyond Cost Savings

Reduced Reliance on On-Site Personnel

- UAVs allow for remote infrastructure monitoring, significantly reducing the need for staff to be physically present in hazardous or hard-to-reach locations.
- This is particularly relevant during crises like pandemics, where limiting personnel exposure is essential.

Enhanced Safety for Workers

- Aerial UAVs eliminate the need for manual inspections in high-risk areas, such as elevated pipeline structures or remote water facilities.
- Indoor UAVs for sewer inspections prevent workers from entering confined spaces with potential exposure to toxic gases or structural hazards.

Faster Inspection Processes

- Aerial UAVs provide large-area coverage in minimal time, allowing for quick assessments of critical water distribution networks.
- Indoor UAVs navigate wastewater and drainage systems autonomously, detecting issues such as cracks, blockages, or leaks in real-time.
- The AI-powered anomaly detection system accelerates data processing, allowing for immediate response to potential infrastructure failures.

Multiple Inspections Without Service Interruptions

- Traditional inspections often require planned shutdowns, whereas UAV inspections can be performed without disrupting operations.
- Routine monitoring can be conducted at higher frequencies without additional costs, enabling a proactive rather than reactive maintenance strategy.

Improved Environmental and Sustainability Performance

- UAV inspections reduce vehicle-based site visits, lowering fuel consumption and carbon footprint.
- Leak detection and pipeline condition monitoring help prevent water loss and ensure a more sustainable resource management strategy.

Enhanced Data Collection and Decision-Making

- The integration of high-resolution imaging, thermal cameras, and AI analytics provides detailed insights into infrastructure conditions.
- The data collected is stored digitally, allowing for historical comparisons and predictive maintenance planning.

The SUNRISE RII Tool represents a transformative shift in how ACOSOL monitors and maintains its water infrastructure. By combining aerial and indoor UAV inspections with advanced AI analysis, ACOSOL can achieve greater efficiency, improved safety, and enhanced resilience in its operations.

6 Conclusions

This deliverable represents the final development stage of the SUNRISE Remote Infrastructure Inspection (RII) Tool within WP7. It brings together a suite of innovative components – including satellite imagery analysis and UAV-based inspection modules, all supported with the artificial intelligence algorithms – into a unified system tailored to the needs of critical infrastructure operators. Furthermore, the final development of the tool is designed with several real-world considerations in mind:

- It is easily portable and can be configured to operate at CI premises.
- Communication with CIs was maintained throughout the development process to ensure that their actual needs were understood and addressed, rather than merely following the initial project plan.
- Integration of legacy data sources into the GUI is possible, allowing CIs to enhance their operations and connectivity.

The RII tool has been carefully designed to enhance infrastructure resilience and support operational continuity in scenarios where access to physical sites is restricted or human resources are limited, such as during pandemics or other temporary disruptions.

A central focus of this deliverable has been the complete description of the developed tools. This deliverable also provided the various considerations about the tools and the reasons why certain development paths and approaches were not pursued. In this way, this deliverable presents the current real-world evaluation of what is technically possible with the current selection of the piloting CI operators. In Pilot 1, each pilot not only validated the technical capabilities of the RII tool but also provided critical feedback, which directly informed refinements in functionality, usability and deployment strategies.

End-user evaluations consistently highlighted the benefits of the tool in improving visibility over hard-to-reach assets, accelerating fault detection and enabling more efficient resource allocation. Despite differences in their operational environments, all pilot partners acknowledged that the RII tool enhances their inspection capabilities and supports continuity planning during personnel shortages.

In conclusion, WP7 has delivered a robust and forward-looking infrastructure inspection solution that meets the SUNRISE vision of improving the resilience of Europe's critical services. The RII tool combines technological maturity with practical applicability and demonstrates that remote inspection is not only a future objective but a current necessity. The SUNRISE project paved the way for broader adoption, while also identifying pathways for continued evolution in areas such as automation, regulatory integration and cross-sector scalability.

References

[1] **Tan, M., & Le, Q. (2019, May)**. Efficientnet: Rethinking model scaling for convolutional neural networks. In International conference on machine learning (pp. 6105-6114). PMLR.

[2] **Ronneberger, O., Fischer, P., & Brox, T. (2015)** U-net: Convolutional networks for biomedical image segmentation, Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18, pp. 234-241.

[3] **Zavrtanik, V., Kristan, M., & Skočaj, D. (2021)**. Reconstruction by inpainting for visual anomaly detection. Pattern Recognition, 112, 107706.

[4] **Lin, T. Y., Dollár, P., Girshick, R., He, K., Hariharan, B., & Belongie, S. (2017)**. Feature pyramid networks for object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 2117-2125).

[5] **Zhao, H., Shi, J., Qi, X., Wang, X., & Jia, J. (2017)**. Pyramid scene parsing network. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 2881-2890).

[6] **Toker, A., Kondmann, L., Weber, M., Eisenberger, M., Camero, A., Hu, J., ... & Leal-Taixé, L. (2022)**, Dynamicearthnet: Daily multi-spectral satellite dataset for semantic change segmentation. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 21158-21167.

[7] **Gupta, R., Goodman, B., Patel, N., Hosfelt, R., Sajeev, S., Heim, E., ... & Gaston, M. (2019)**, Creating xBD: A dataset for assessing building damage from satellite imagery, *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops* (pp. 10-17).

[8] **Chen, H., & Shi, Z. (2020)**, A spatial-temporal attention-based method and a new dataset for remote sensing image change detection, *Remote Sensing*, 1662.

[9] **Fang, S., Li, K., & Li, Z. (2023)**, Changer: Feature interaction is what you need for change detection, *IEEE Transactions on Geoscience and Remote Sensing.*

[10] **Chen, H., Qi, Z., & Shi, Z. (2021)**, Remote sensing image change detection with Transformers, IEEE Transactions on Geoscience and Remote Sensing, 60, 1-14.

[11] **Çiçek, Ö., Abdulkadir, A., Lienkamp, S. S., Brox, T., & Ronneberger, O. (2016)**. 3D U-Net: learning dense volumetric segmentation from sparse annotation. In Medical Image Computing and Computer-Assisted Intervention–MICCAI 2016: 19th International Conference, Athens, Greece, October 17-21, 2016, Proceedings, Part II 19 (pp. 424-432). Springer International Publishing.

[12] **Garnot, V. S. F., & Landrieu, L. (2021)**, Panoptic segmentation of satellite image time series with convolutional temporal attention networks, *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4872-4881.

[13] **Noh, H., Ju, J., Seo, M., Park, J., & Choi, D. G. (2022)**, Unsupervised change detection based on image reconstruction loss, *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1352-1361.

[14] **Rombach, R., Blattmann, A., Lorenz, D., Esser, P., & Ommer, B. (2022)**. High-resolution image synthesis with latent diffusion models. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 10684-10695).

[15] **Xu, J., Liu, S., Vahdat, A., Byeon, W., Wang, X., & De Mello, S. (2023)**. Open-vocabulary panoptic segmentation with text-to-image diffusion models. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 2955-2966).

[16] **Ginzler, Christian (2021)**, Vegetation Height Model NFI. National Forest Inventory (NFI), doi:10.16904/1000001.1.

[17] **Chen, L. C., Papandreou, G., Schroff, F., & Adam, H. (2017)**, Rethinking atrous convolution for semantic image segmentation, *arXiv preprint arXiv:1706.05587*.

[18] **Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., ... & Guo, B. (2021)**, Swin transformer: Hierarchical vision transformer using shifted windows, *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 10012-10022.

[19] **Liu, Z., Mao, H., Wu, C. Y., Feichtenhofer, C., Darrell, T., & Xie, S. (2022)**, A convnet for the 2020s. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition,* pp. 11976-11986.

[20] **Roboflow Universe**, Crack Dataset by University, https://universe.roboflow.com/university-bswxt/crack-bphdr, retrieved 2024-03-22.

[21] **Roboflow Universe**, Crack Detection Using Instance Segmentation in YOLOv8 by Bach Khoa Ho Chi Minh University, https://universe.roboflow.com/bach-khoa-ho-chi-minh-university-fyr43/crack-detection-using-instance-segmentation-in-yolov8, retrieved 2024-03-22.

[22] **Roboflow**, Corrosion Dataset, https://app.roboflow.com/project/corrosion-jzy5h/1, retrieved 2024-03-22.

[23] **Liu, S., Zeng, Z., Ren, T., Li, F., Zhang, H., Yang, J., Li, C., Yang, J., Su, H., Zhu, J. & Zhang, L., 2023**. Grounding DINO: Marrying DINO with Grounded Pre-Training for Open-Set Object Detection. arXiv:2303.05499v4 [cs.CV], [online] Available at: https://arxiv.org/pdf/2303.05499.pdf, retrieved 2024-04-02.

[24] **Ke, L., Ye, M., Danelljan, M., Liu, Y., Tang, C-K., Yu, F., Tai, Y-W., 2023**. Segment Anything in High Quality. ETH Zürich & HKUST. Available at: https://arxiv.org/pdf/2306.01567.pdf, retrieved 2024-04-02.

[25] **Ultralytics, 2023**. Ultralytics GitHub Repository. [online] Available at: https://github.com/ultralytics/ultralytics, retrieved 2024-04-02.

[26] **Cheng, T., Song, L., Ge, Y., Liu, W., Wang, X., & Shan, Y., 2023.** YOLO-World: Real-Time Open-Vocabulary Object Detection. arXiv:2401.17270 [cs.CV], [online] Available at: https://arxiv.org/abs/2401.17270, retrieved 2024-04-09.

[27] **Zhou, X., Girdhar, R., Joulin, A., Krähenbühl, P. & Misra, I., 2022.** Detecting Twenty-thousand Classes using Image-level Supervision. arXiv:2201.02605v3 [cs.CV]. Available at: https://arxiv.org/pdf/2201.02605.pdf, retrieved 2024-04-09.

[28] **Zou, X., Dou, Z-Y., Yang, J., Gan, Z., Li, L., Li, C., Dai, X., Behl, H., Wang, J., Yuan, L., Peng, N., Wang, L., Lee, Y.J. & Gao, J., 2022**. Generalized Decoding for Pixel, Image, and Language. University of Wisconsin-Madison, UCLA & Microsoft Research at Redmond. Available at: https://arxiv.org/pdf/2212.11270.pdf, retrieved 2024-04-09.

[29] **Li, J., Li, D., Savarese, S. & Hoi, S., 2023**. BLIP-2: Bootstrapping Language-Image Pre-training with Frozen Image Encoders and Large Language Models. Salesforce Research. Available at: https://arxiv.org/pdf/2301.12597.pdf, retrieved 2024-04-09.

[30] **Wu, P. & Xie, S., 2023**. V*: Guided Visual Search as a Core Mechanism in Multimodal LLMs. arXiv:2312.14135 [cs.CV], [online] Available at: https://arxiv.org/abs/2312.14135, retrieved 2024-04-02.

[31] **LLaVA Team. (2024, January 30**). LLaVA: The Next Chapter. Retrieved from https://llava-vl.github.io/blog/2024-01-30-llava-next/, retrieved 2024-04-04.

[32] **Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. (2023).** Improved Baselines with Visual Instruction Tuning. arXiv:2310.03744. Available at: https://arxiv.org/abs/2310.03744, retrieved 2024-04-11.

[33] **Abdin, M., Aneja, J., Awadalla, H., Awadallah, A., et al. (2024**). Phi-3 Technical Report: A Highly Capable Language Model Locally on Your Phone. arXiv preprint arXiv:2404.14219. Available at: https://arxiv.org/abs/2404.14219, retrieved 2025-03-15.

[34] **Gunasekar, S., Gupta, V., Liu, C., Aneja, J., et al. (2025**). Phi-4: A Multi-Modal Intelligence Model for Text, Vision, and Speech. arXiv preprint arXiv:2503.01743. Available at: https://arxiv.org/abs/2503.01743, retrieved 2025-03-15.

[35] **Bai, Y., Yang, S., Li, G., Chen, X., et al. (2024)**. Molmo: Rethinking the Role of Mixture-of-Experts in Multimodal Large Language Models. arXiv preprint arXiv:2409.17146. Available at: https://arxiv.org/abs/2409.17146, retrieved 2025-03-15.

[36] **Yang, J., Zhang, B., Yang, B., Hui, B., et al. (2025**). Qwen-VL: A Frontier Large Vision-Language Model with Versatile Multimodal Capabilities. arXiv preprint arXiv:2502.13923. Available at: https://arxiv.org/abs/2502.13923, retrieved 2025-03-15.

[37] **Zhu, X., et al., 2020.** TinaFace: Strong but Simple Baseline for Face Detection. arXiv preprint arXiv:2011.13183v3. [online] Available at: https://arxiv.org/pdf/2011.13183v3.pdf, accessed 2024-04-04.

[38] Sunrise Project. (2023, December 11). **SUNRISE Tool for Remote Infrastructure Inspection Demo**. [Video]. YouTube. https://youtu.be/2zYAEmabfKs, retrieved 2024-04-11.

[39] **Papers with Code**, 2024. Face Detection Task Overview. [online] Available at: https://paperswithcode.com/task/face-detection, retrieved 2024-04-04.

[40] **Media-Smart**, 2024. TinaFace Configuration in Vedadet Repository. [online] Available at: https://github.com/Media-Smart/vedadet/tree/main/configs/trainval/tinaface, retrieved 2024-04-04.

[41] **OpenAI (2023)**. GPT-4 Technical Report. Available at: https://arxiv.org/pdf/2303.08774.pdf, retrieved 2024-04-04.

[42] **ZeroTier, 2024**. ZeroTier: Networks. [online] Available at: https://my.zerotier.com/, retrieved 2024-04-04.

[43] Republic of Slovenia, '**Decree on implementing Regulation (EU) on the rules and procedures for the operation of unmanned aircraft**' (Official Gazette of the Republic of Slovenia, no. 195/20 and no. 31/21). [online] Available at: https://pisrs.si/pregledPredpisa?id=URED8075 , retrieved 2024-03-22.

[44] European Commission, '**COMMISSION IMPLEMENTING REGULATION (EU) 2019/947 of 24 May 2019 on the rules and procedures for the operation of unmanned aircraft**'. [online] Available at: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A02019R0947-20220404 , retrieved 2024-03-22.

[45] **SUNRISE. D7.1** Infrastructure inspection conceptualization. Dejan Štepec. 2023.

[46] **SUNRISE. D7.2** Infrastructure inspection tool and training guide V1. Mario Triviño. 2023

[47] **SUNRISE. D7.3** Infrastructure inspection tool and training guide V2. George Leventakis. 2024

[48] **SUNRISE. D7.4** Infrastructure inspection pilot report V1. Milan Tarman. 2024

[49] Sunrise Project. (2024, July 30). **SUNRISE RII UAV rea-time fire detection**. [Video]. YouTube. https://youtu.be/73gl528cJ88, retrieved 2025-03-11.

[50] Sunrise Project. (2025, February 17). **SUNRISE Remote Infrastructure Inspection Tool Piloting Work**. [Video]. YouTube. https://youtu.be/p6W6KGuqkww, retrieved 2025-03-11.

[51] https://www.e-plaz.si/

[52] https://mlflow.org/

[53] Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R. and Ng, R., 2020. NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis. UC Berkeley, Google Research, UC San Diego. Available at: https://arxiv.org/pdf/2003.08934.pdf

[54] Müller, T., Evans, A., Schied, C., & Keller, A., 2022. Instant Neural Graphics Primitives with a Multiresolution Hash Encoding. NVIDIA. Available at: https://nvlabs.github.io/instant-ngp/assets/mueller2022instant.pdf

[55] Schönberger, J.L., Frahm, J-M., 2016. Structure-from-Motion Revisited. University of North Carolina at Chapel Hill & Eidgenössische Technische Hochschule Zürich. Available at: https://demuc.de/papers/schoenberger2016sfm.pdf

[56] **HuggingFace** (2025). Microsoft Phi-4-multimodal-instruct model summary. [online] Available at: microsoft/Phi-4-multimodal-instruct · Hugging Face, retrieved 2025-03-11.

[57] Sunrise Project. (2024, November 22). **SUNRISE UAV Remote Inspection Tool. UAV GUI configurate new use case**. [Video]. YouTube. https://youtu.be/p6W6KGuqkww, retrieved 2025-03-11.

[58] Inpaint Anything https://github.com/geekyutao/Inpaint-Anything

[59] Sentinel Hub www.sentinel-hub.com

[60] SkyFi https://skyfi.com

[61] https://www.whiteboxgeo.com/whitebox-workflows-for-python/

[62] Docker https://www.docker.com/

[63] Open Stack https://www.openstack.org/

[64] FastAPI https://fastapi.tiangolo.com/

Annex I: Satellite inspection API User Guide

This section provides a user guide for the API system of the Satellite Inspection tool. The system includes two submodules, namely vegetation height and change detection, at two different endpoints. Before submitting the request, we first need to define the geographical feature describing the location where we want to perform the satellite inspection. Such features should be submitted to the API in a GeoJSON format[1]. The job can be posted to the appropriate endpoint (*/{sub-module}/prediction* where the sub-module can be any of vegetation-height or change-detection) using curl command or interactively with API docs as shown in Figure 49 and Figure 50. Currently, our module supports Point and Polygon geographical features. After submitting the job, the application responds with the job_id and starts computing the output in the background. The job_id can then be used to check the status of the job and receive the output.

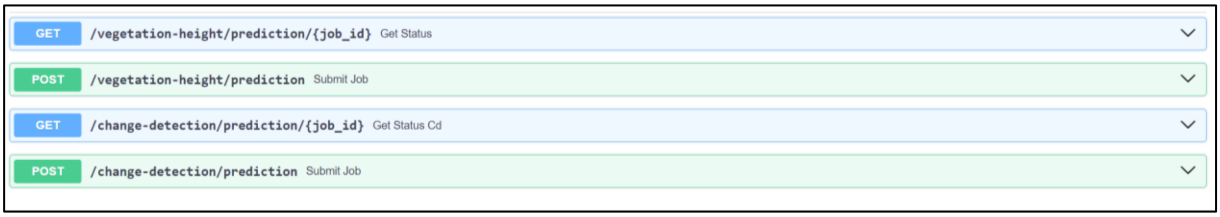

Figure 49: Endpoints of the satellite inspection API for posting jobs and receiving results.

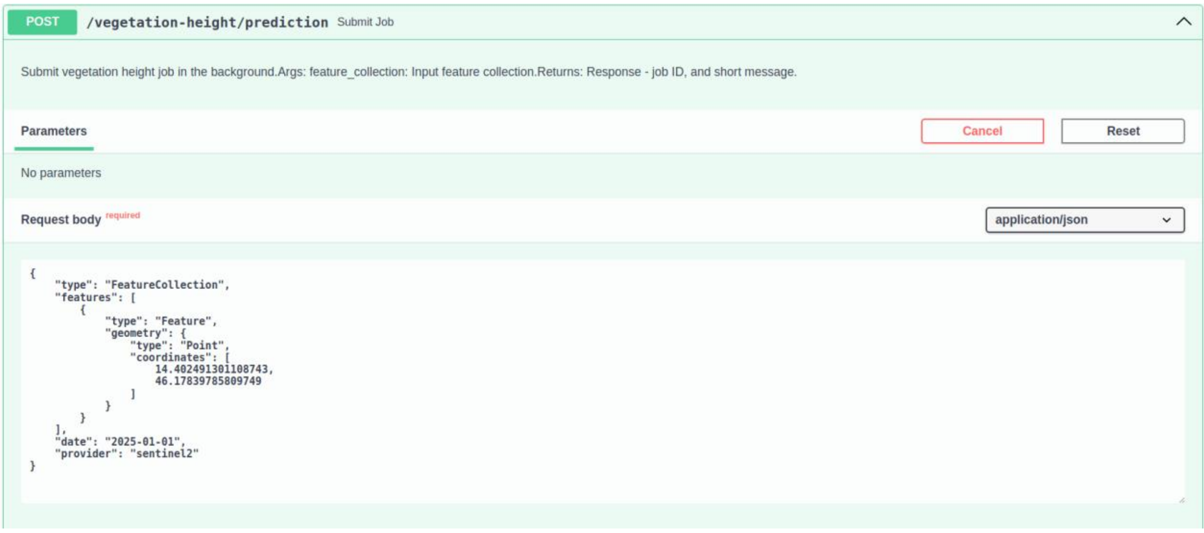

Figure 50: Example of posting a job to vegetation-height submodule with a point geographical feature using Sentinel2 satellite image provider.

[1] https://geojson.org/

To obtain the status and result, send a GET request at */{sub_module}/prediction/{job_id}*. Possible statuses are listed in Table 10. The Full response Schema is shown in Figure 51 and is described in further detail in Table 11.

Table 10: Possible statuses of the job submitted to the satellite inspection module.

Status
Job_acepted
In_progress
Searching_satellite_images
Downloading_satelite_images
Cropping_satellite_images
Running_inference
Done
Image_not_found

Table 11: Description of response fields.

Field		Description
"status"		Status.
"history"		Previous statuses with timestamps.
"messages"	"header"	Basic info.
	"media"	Satellite image, encoded in base64.
	"detection"	Detection output. Result is given in GeoJSON format. Vegetation height module returns polygons with specific height of vegetation. Change detection module returns polygons where the change was detected.

```
{
  "status": "job_accepted",
  "history": [
    {
      "status": "job_accepted",
      "timestamp": 0
    }
  ],
  "messages": [
    {
      "header": {
        "source-type": "SAT",
        "source-id": "XLAB",
        "message-uuid": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
        "message-timestamp": 0
      },
      "media": [
        {
          "media-uuid": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
          "mime-type": "image/jpeg",
          "timestamp": 0,
          "data-base64": "string",
          "remote-url": "string",
          "geo-reference": {
            "latitude": 0,
            "longitude": 0
          }
        }
      ],
      "detection": {
        "detection-uuid": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
        "source-timestamp": 0,
        "processed-timestamp": {
          "start": 0,
          "end": 0
        },
        "geo-reference": {
          "latitude": 0,
          "longitude": 0
        },
        "class-level": {
          "label": "string",
          "confidence": 0
        },
        "detection-label": "string",
        "detection-description": "string",
        "geoJson": {
          "type": "FeatureCollection",
          "features": [
            {
              "type": "Point",
              "coordinates": [null, null]
            },
            {
              "type": "Polygon",
              "coordinates": [
                [
                  [null, null]
                ]
              ]
            }
          ]
        }
      }
    }
  ]
}
```


Figure 51: JSON response schema / example value.

Examples of CURLs:

- Request vegetation height prediction at the given geographic coordinate:

```
curl -X 'POST' \
 'https://remote-inspection.xlab.si/vegetation-height/prediction' \
 -H 'accept: application/json' \
 -H 'Content-Type: application/json' \
 -d '{
  "type": "FeatureCollection",
  "features": [
    {
      "type": "Feature",
      "geometry": {
        "type": "Point",
        "coordinates": [
          14.402491301108743,
          46.17839785809749
```

```
        ]
      }
    }
  ],
  "date": "2023-10-01",
  "provider": "sentinel2"
}'
```


Response body


```
{
 "msg": "Prediction job accepted. Send GET to /vegetation-height/prediction to
inspect the results",
 "job_id": "ce02b39b-f425-435c-ac67-06ee32147f4a"
}
```


- Check status of the job

```
curl -X 'GET' \
 'https://remote-inspection.xlab.si/vegetation-height/prediction/ce02b39b-f425-435c-ac67-06ee32147f4a' \
 -H 'accept: application/json'
```

Response body (note that it is heavily cut).


```
{
 "status": "done",
 "history": [
  {
   "status": "job_accepted",
   "timestamp": 1742562116063
  },
  {
   "status": "in_progress",
   "timestamp": 1742562116066
  },
  ...
  },
  {
   "status": "running_inference",
   "timestamp": 1742562147829
  },
  {
   "status": "done",
   "timestamp": 1742562150567
  }
 ],
 "messages": [
  {
   "header": {
    "source-type": "SAT",
    "source-id": "XLAB",
    "message-uuid": "ce02b39b-f425-435c-ac67-06ee32147f4a",
    "message-timestamp": 1742562150580
   },
   "media": [
    {
     "media-uuid": "bf5947f8-0654-11f0-8b40-0242ac110002",
     "mime-type": "image/jpeg",
     "timestamp": 1695808831000,
     "data-base64": "...",
     "remote-url": null,
     "geo-reference": null
    }
   ],
   "detection": {
```

```
        "detection-uuid": "bf596238-0654-11f0-8b40-0242ac110002",
        "source-timestamp": 1742562140580,
        "processed-timestamp": {
         "start": 1742562150580,
         "end": 1742562160580
        },
        "geo-reference": {
         "latitude": 46.17839785809749,
         "longitude": 14.402491301108743
        },
        "class-level": {
         "label": null,
         "confidence": null
        },
        "detection-label": null,
        "detection-description": null,
        "geoJson": {
         "type": "FeatureCollection",
         "features": [
          {
           "type": "Feature",
           "geometry": {
            "type": "Polygon",
            "coordinates": [
             [
              [
               14.395923444626035,
               46.177755573449055
              ],
              [
               14.395924430143479,
               46.17766557782458
              ],
              [
               14.396054968584348,
               46.17757626672289
              ],
              [
               14.396052997976819,
               46.17775625797755
              ],
              [
               14.395923444626035,
               46.177755573449055
              ]
             ]
            ]
           },
           "properties": {
            "height": "min=0_max=5"
           }
          },
          ...
         ]
        }
       }
      }
     ]
}
```

Annex II: UAV Image Processing API System User Guide

This second annex of the document introduces the final version of the **User Guide** for Critical Infrastructure (CI) end-users. As previously specified, the intended user profile corresponds to personnel responsible for overseeing maintenance activities in CI facilities. This includes individuals with a technical background but without the need for extensive knowledge in software development, artificial intelligence, computer vision, or other highly specialized fields.

To ensure seamless adoption, multiple **training sessions** have been conducted using the **inspection tool web interface** across all WP7 CIs. These sessions provided private access to the tool, allowing stakeholders to explore its functionalities. Notably, HDE has demonstrated significant interest in customizing AI pipelines for their specific use cases. The tests confirmed that the current interface is clear and intuitive, enabling the designated users to construct dynamic inspection pipelines that add substantial value to their maintenance and risk prevention operations.

In this final version of the tool, users can interact in two distinct ways:

- **Through the GUI (Graphical User Interface):**
  - Designed, developed, and deployed to maximize accessibility, making the tool usable across diverse user profiles.
  - Eliminates the need for command-line interactions, simplifying the creation and execution of new inspection pipelines.
  - Supports preconfigured scenario loading, requiring only basic knowledge of the web interface.
- **Through direct API requests (via CURL or other HTTP request tools):**
  - All available API parameters have been documented to facilitate usage.
  - A **virtual assistant powered by ChatGPT** has been integrated to assist users in crafting complex queries based on **natural language descriptions** or **reference images** related to the inspection problem at hand.

Due to these interaction modalities, this annex is divided into two sections:

- **Part A** focuses on the UAV **GUI-based** inspection tool.
- **Part B** details the **virtual assistant** and the **API request definitions** for command-line interaction.

While both options are available, the UAV GUI is the recommended approach, as it has been specifically deployed to simplify API interactions and ensure that the UAV Remote Inspection Tool remains highly **user-friendly**.

Regardless of the chosen interaction mode, whether through the GUI or direct API requests, the first requirement is to obtain access to the private VPN established to ensure secure and exclusive remote connectivity to the APIs and GUI. This VPN access, implemented through a **ZeroTier private network**, is restricted to WP7 Critical Infrastructures (CIs). Authorization must be requested from ATS before use.

Annex II: Part A – UAV GUI

Once the end-user has been authorized in the ZeroTier private network, they will be able to access the GUI through their preferred web browser using the address http://{server}:{port}, which will be provided upon registration.

Before proceeding with the detailed user guide, it is highly recommended to watch the **demonstration video** [57]. This video provides a step-by-step walkthrough of how to configure the tool for a specific use case, offering users a comprehensive overview of the available options and functionalities before diving deeper into the guide.

Once the user is familiar with the UAV GUI, they can enter the provided URL into their browser. After loading the page, the home screen will appear, as illustrated in Figure 52, with the "Batch Processing" tab selected by default.

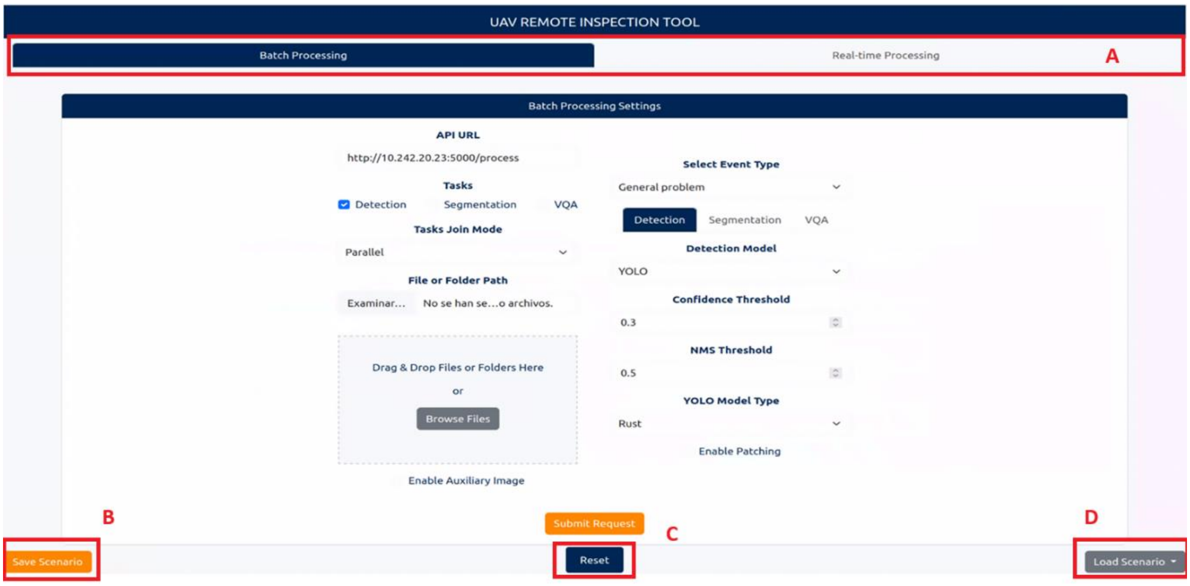

Figure 52: Home page with tab “Batch Processing” selected. A- Analysis mode tab selector; B- Save settings button; C- Reset settings button; D- Load settings button.

Regarding this main window, four common elements remain constant across both tabs of the interface: “Batch Processing” and “Real-Time Processing”. These elements are highlighted with red boxes in Figure 52 and labeled A, B, C, and D for identification.

- **A** marks the tab selector, which allows the user to choose between real-time processing and batch processing on previously collected images. The user simply needs to click on the desired tab to switch to the specific menu for that mode.
- **B** corresponds to the "Save Scenario" button, which enables the user to save an inspection scenario, i.e., a complete set of parameters for a specific use case. The user must define a name for this parameter set.
- **C** is the "Reset" button, which restores all values to their default settings, allowing the user to clear any configurations made during the session.
- **D** represents the "Load Scenario" button, which allows loading a previously saved parameter set, either one saved manually by the user (via button B) or one of the default presets configured for key inspection scenarios, such as rust detection, ceramic insulator inspection, flood detection, etc.

If the user selects the **real-time processing tab** using the selector in A, the menu shown in Figure 53 will be displayed.

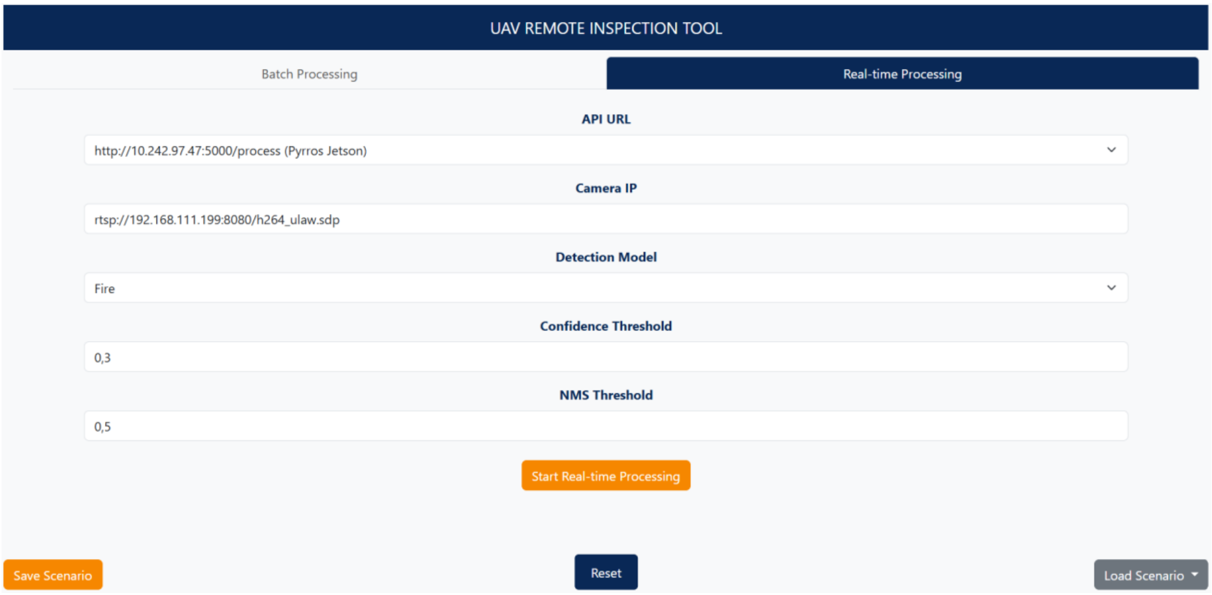

Figure 53: Home page with tab "Real-time Processing" selected.

As can be observed, the real-time processing mode is significantly less complex than batch processing, offering a smaller number of configurable parameters. This menu allows the user to specify the API URL, which determines where the real-time video analysis API is running. The location of the API will depend on the hardware setup, as it can be deployed on the Edge (Jetson device) or in the cloud on the server. For this reason, the API URL is presented as a dropdown menu, enabling the user to select which API instance to connect to.

The second parameter is the camera IP address, which represents the video stream source to be analyzed. This value depends on the UAV transmitting the footage and must be configured accordingly to ensure proper data capture.

The third configurable setting is the detection model, allowing users to select which AI model will be applied to the real-time video stream. Within this project, three pretrained real-time models have been developed: Fire Detection, Rust Detection, and Crack Detection, as illustrated in Figure 54.

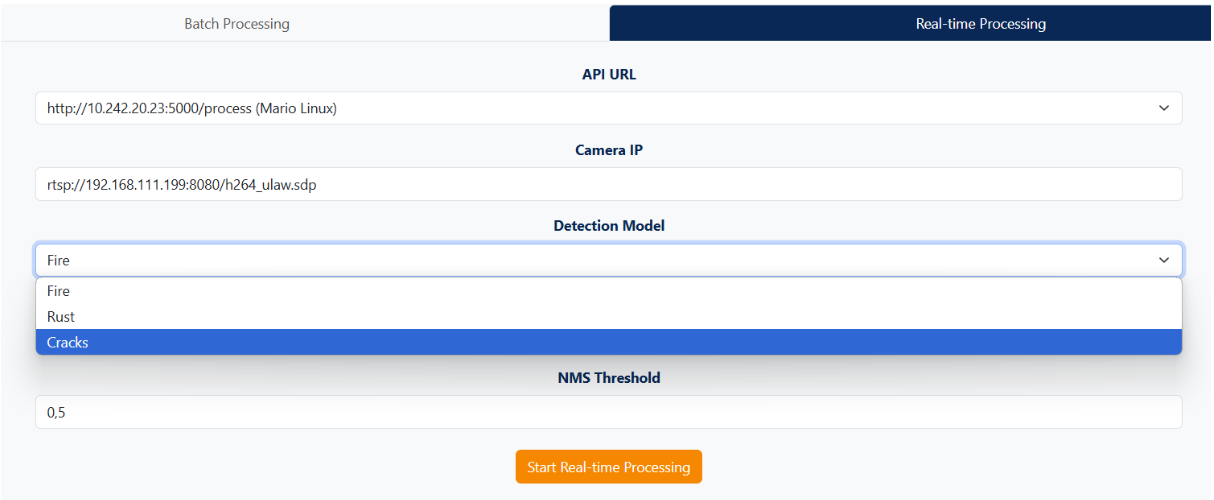

Figure 54: "Real-time Processing" tab configuration settings.

The last two parameters available in the real-time processing tab, Figure 53, are Confidence Threshold and Non-Maximum Suppression (NMS), which allow the user to fine-tune the detection performance. Confidence Threshold defines the minimum probability required for an object to be classified as a detection, filtering out low-confidence results. NMS (Non-Maximum Suppression) helps reduce redundant overlapping detections, ensuring that only the most relevant bounding boxes are retained.

By adjusting these values, the user can balance accuracy and false positives, optimizing the inspection process according to the specific operational requirements.

Once these few parameters have been configured, the CI operator can begin analyzing the images streamed by the UAV in real time by clicking the "Start Real-time Processing" button located at the bottom of the screen.

After briefly defining the real-time processing tab, let's now focus on the more powerful and flexible **batch processing tab**. The user simply needs to switch back to this tab using the tab selector, which was previously defined and highlighted as A in Figure 52. Upon doing so, they will see the menu structured into two columns, as illustrated in Figure 55.

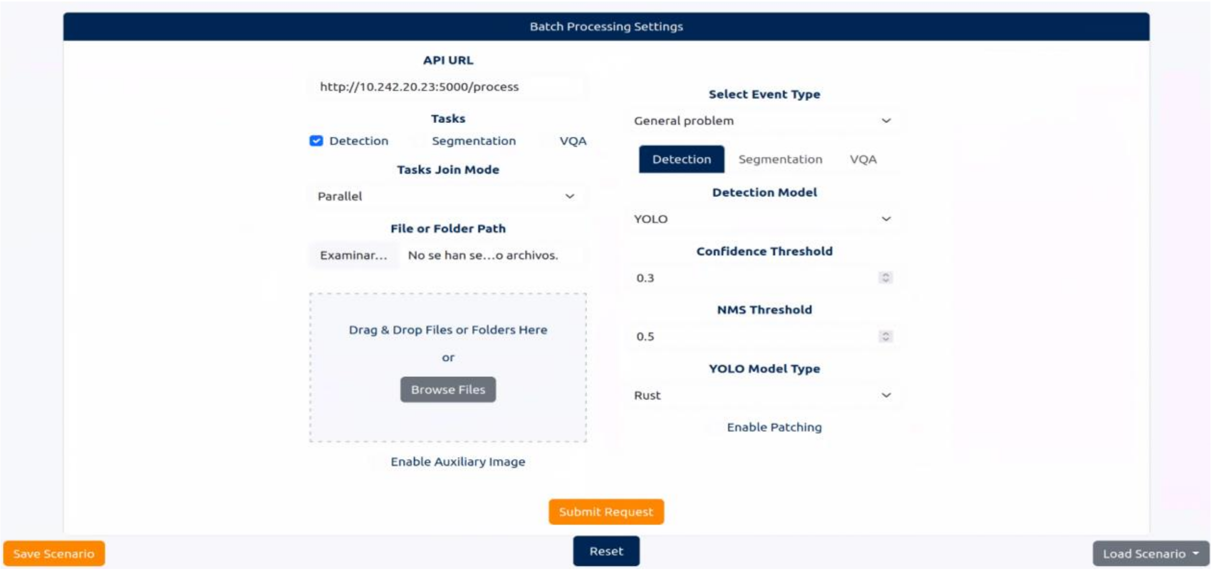

Figure 55: "Batch Processing" tab main configuration settings.

The left column of the Batch Processing tab menu contains the essential elements needed to define a pipeline and send a request to the API, while the right column allows users to fine-tune the specific AI models they wish to use.

In the left column, at the top, the first element is the API URL field, which, just like in the real-time tab, enables the user to define the server address where AI analysis requests will be sent.

Directly below, there are three checkboxes that allow the user to select which tasks to apply: detection, segmentation, and VQA. If a task is not selected, any parameters set in the corresponding subtabs of the right column will be ignored, as that processing module will not be applied to the image under analysis.

Once the tasks have been selected, the end-user can define whether they should be executed in parallel or in cascade using the Task Join Mode dropdown menu.

- If parallel is selected, each module operates independently, meaning the output of one does not influence the others.
- If cascade is selected, the output of the segmentation module is used as the input for detection and VQA (if those tasks are enabled).

Beyond these pipeline configuration parameters, the left column includes two fields for selecting the image or video to be processed, either by browsing local files or using the drag-and-drop functionality.

If a video file is selected instead of an image, an additional field appears, allowing the user to define the frames per second (FPS) to be processed from the video. This is illustrated in Figure 56, which isolates the left column of the interface for clarity.

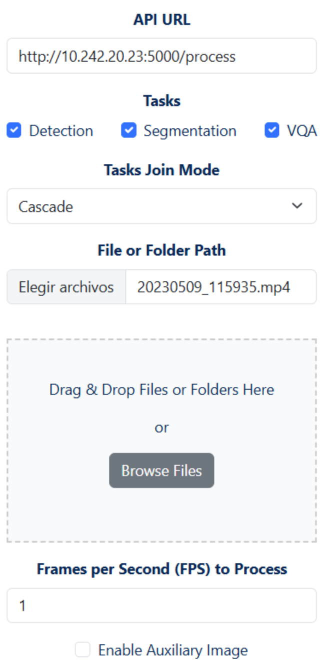

Figure 56: “Batch Processing” tab, video FPS to process field.

Additionally, for use cases requiring multiple input images, such as comparing two images using the VQA module or incorporating auxiliary images from different spectra, like infrared (IR), there is an option available at the bottom of the left column, as shown in Figure 57. A checkbox can be selected to enable this functionality, which then displays a second input field where the user can specify the path of the auxiliary image to be used in the analysis.

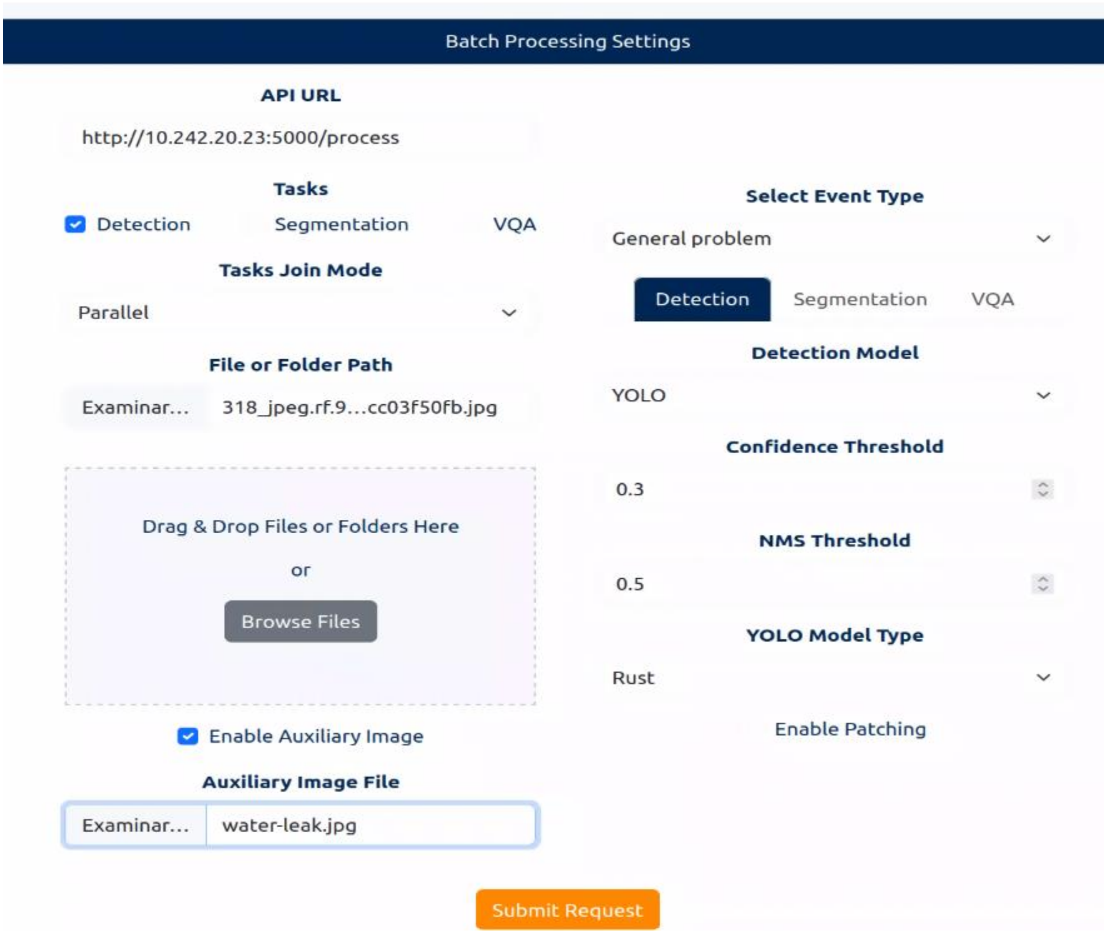

Figure 57. “Batch Processing” tab enable auxiliar image configuration settings selected.

Regarding the **right column of the Batch Processing tab** interface, the first element is a drop-down menu that allows users to select the type of alert that will be sent to the WP7 general GUI if the image analysis determines that the infrastructure has a potential issue. This functionality helps users filter alerts by type within the centralized alert management interface. The system allows defining as many alert types as needed, providing flexibility in categorizing inspection results. Figure 58 illustrates some examples of these alert settings.

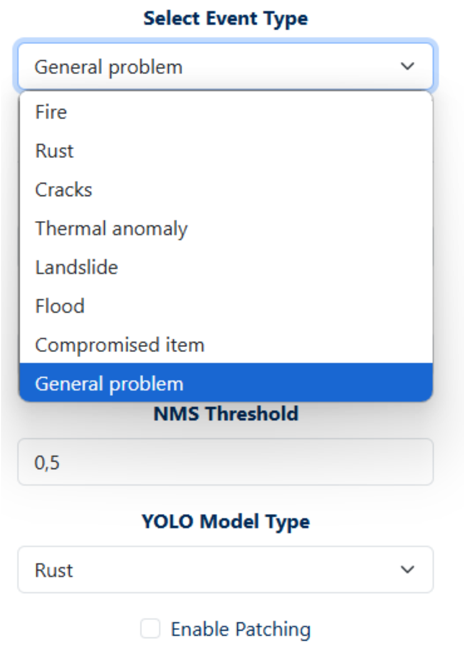

Figure 58: “Batch Processing” tab, “Select Event Type” drop-menu settings.

Below this menu, the subtabs for different processing modules are displayed, allowing the user to navigate between them by simply clicking on the desired tab title. The first subtab corresponds to the **detection module**, where the first field defines which detection model will be used, offering the choice between YOLO and GroundingSAM (grdSAM). If the user selects YOLO, the menu changes to display the necessary settings for this model, as shown in Figure 59.

Among the parameters available in this configuration, the first two are Confidence Threshold and NMS, both previously explained in other sections. Additionally, the user can select the **specific YOLO model** to be applied from the available trained options. Another configurable option is an optional "Patching" checkbox, which allows the user to decide whether to crop the original image into smaller sections before running the detection process. This approach increases the aspect ratio of objects and can improve detection accuracy for small defects or elements within large images. However, it significantly slows down the detection process, as the system must crop the image into patches, run detection separately on each section, and then stitch the results together, applying Non-Maximum Suppression (NMS) to avoid redundant detections.

If the user chooses to enable patching, additional text fields appear, allowing them to specify the patch size and the overlap between consecutive windows, further customizing the detection process according to their specific requirements.

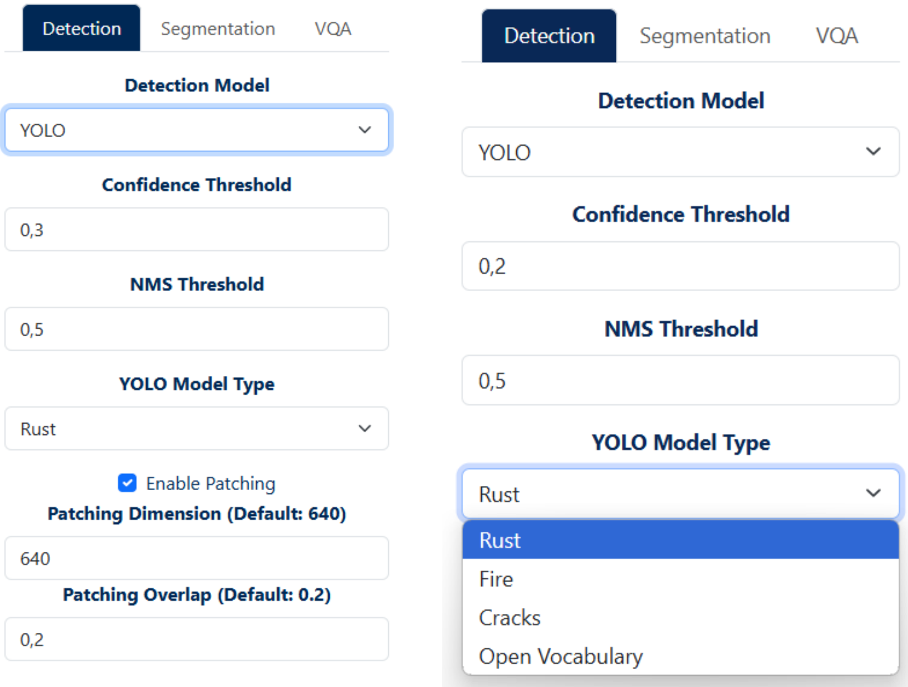

Figure 59: “Batch Processing” tab, “Detection” configuration settings. YOLO model.

If instead of selecting YOLO as the detection model, the user chooses **GroundingSAM**, several configuration fields remain the same, such as confidence threshold, NMS, and the option to enable patching. However, two additional fields appear, allowing the user to specify which classes should be detected by this zero-shot, open-vocabulary detector and which of these should be highlighted in the final output. This menu can be found in Figure 60.

This distinction between detected classes and highlighted classes is crucial, as it helps improve clarity in scenarios where the detection task involves concepts with overlapping semantics. For example, in a case where the user wants to detect clogged grates, simply using the category "clogged grate" might lead to misclassifications, as the detector could assign high confidence scores to any grate, even if it is not actually clogged, since the concept is inherently related to "grate" itself. By defining both "grate" and "clogged grate" as detected classes, the model first ensures that all relevant grates are identified. Then, by selecting only "clogged grate" as the highlighted class, the user can filter the final results to focus only on the specific detections of interest, preventing clean grates from being incorrectly classified as clogged.

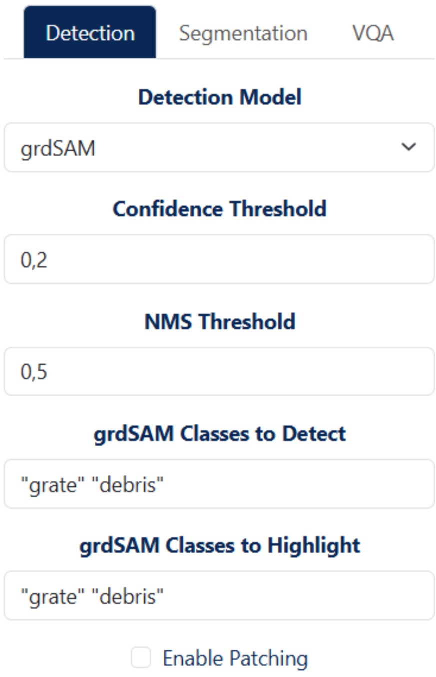

Figure 60: “Batch Processing” tab, “Detection” configuration settings. GrdSAM model.

The second subtab is **Segmentation**, which, as shown in Figure 61, contains configuration fields very similar to those already introduced in the Detection module. This task allows users to choose between GroundingSAM and X-Decoder as the segmentation model.

In most cases, **GroundingSAM** is the preferred option when segmenting a specific and well-defined object or structure, such as a concrete sleeper on railway tracks. On the other hand**, X-Decoder** is more suitable for scene-wide semantic segmentation, where the goal is to classify multiple zone/elements in an image, such as detecting flooded areas over a railway track after heavy rain.

The only parameter not previously explained is Image Size for X-Decoder. It is generally recommended not to increase this value, as higher resolutions can lead to GPU memory limitations. If the image does not contain too many small objects or is not overly complex, reducing this value may improve performance without significantly affecting segmentation quality.

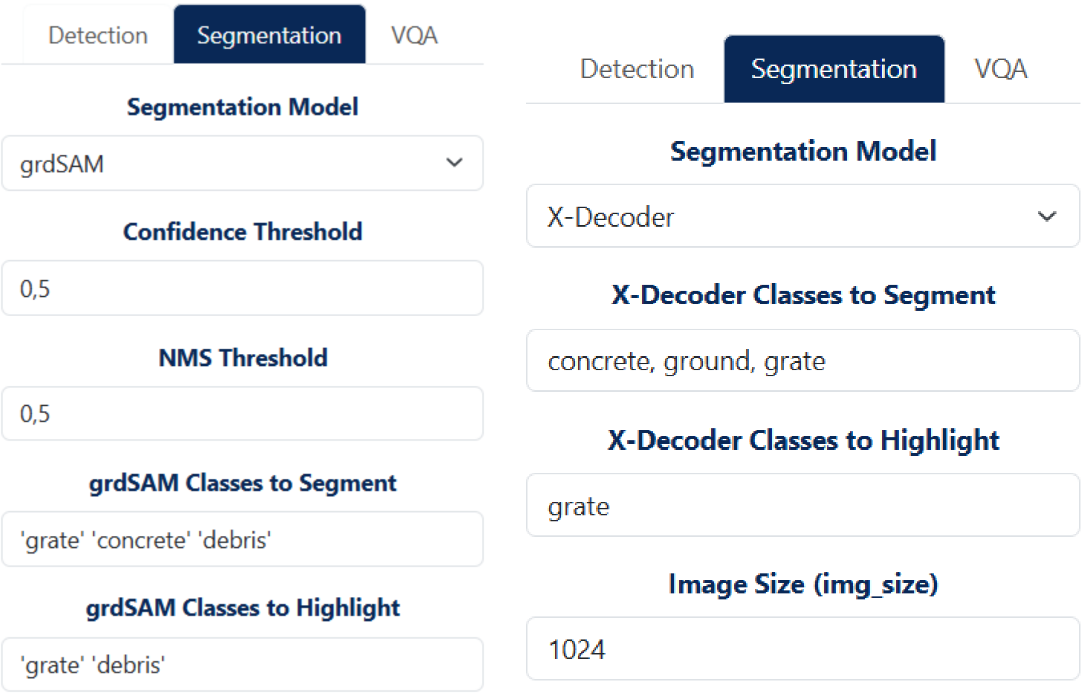

Figure 61: “Batch Processing” tab, “Segmentation” configuration settings.

The last subtab is the **VQA (Visual Question Answering)** module, where the user can define the model to be used, the questions to be asked about the image or images, and the keywords or VQA triggers that will activate a "1" in the status vector whenever they appear in the response generated by the VLM (Vision-Language Model).

Although multiple model options are available, as shown in Figure 62, it is highly recommended to use the Phi family of models in all scenarios. These models significantly outperform previous versions, delivering superior accuracy and reliability in answering visual queries.

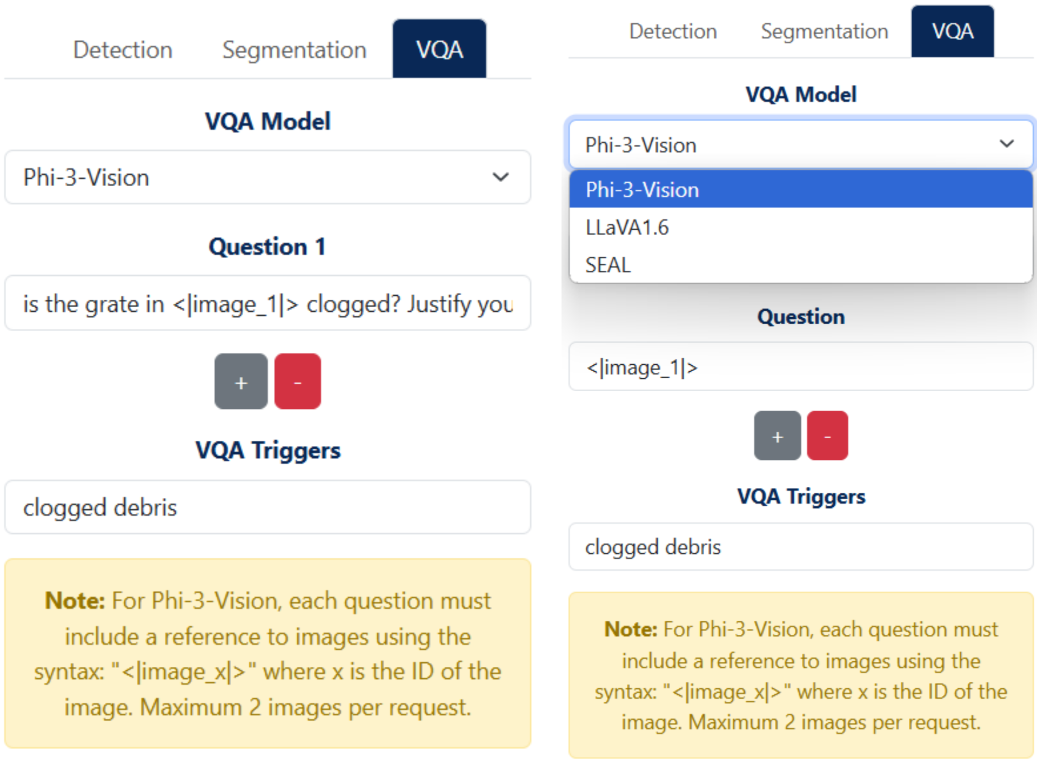

Figure 62: “Batch Processing” tab, “VQA” configuration settings.

After finalizing all the parameter configurations, the user can proceed by clicking the "Submit Request" button, visible in Figure 57, to send the request to the main API of the RII UAV tool. This action will initiate the processing, and a loading screen, similar to the one shown in Figure 63, will be displayed for the duration of the analysis.

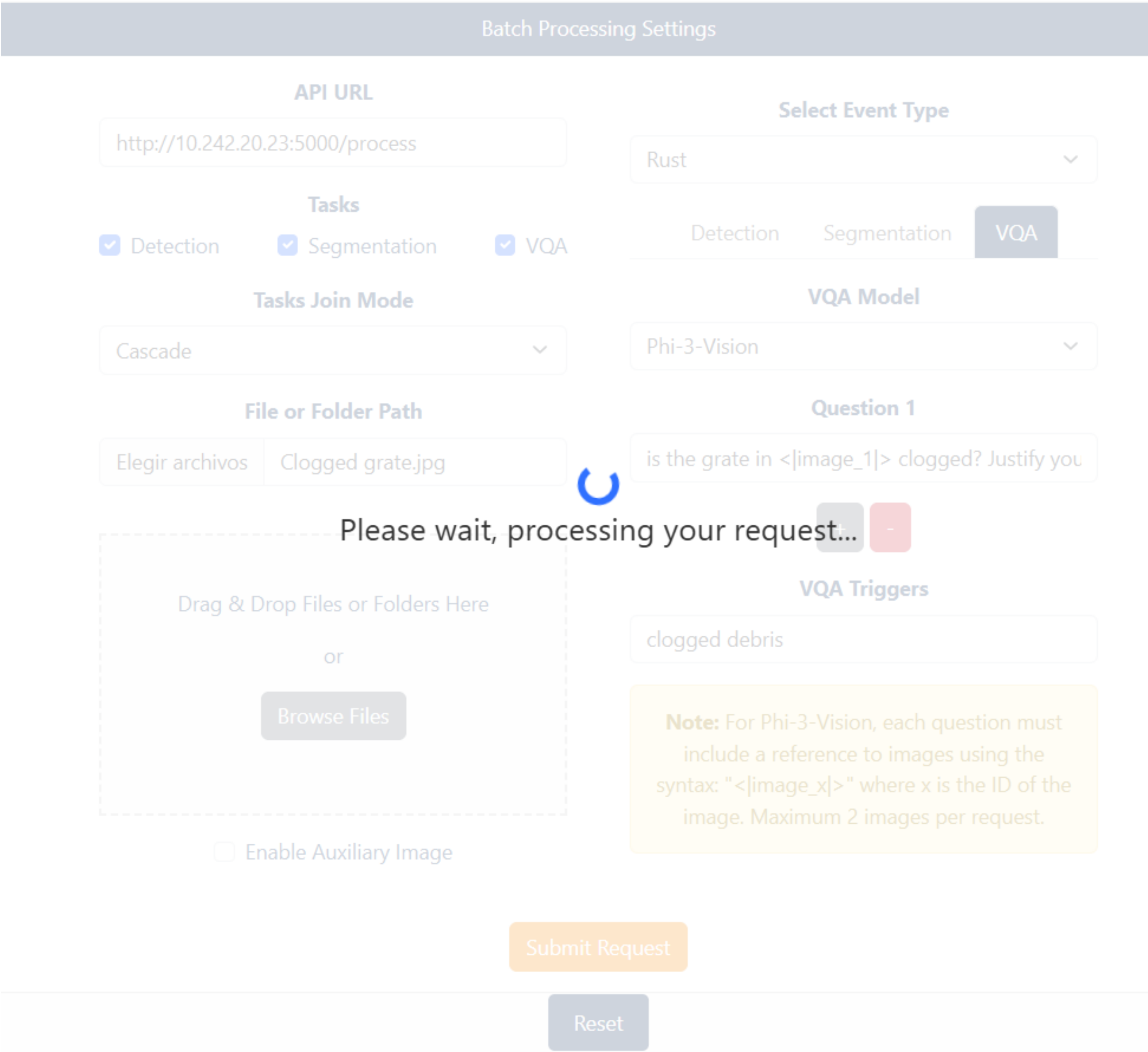

Figure 63: "Batch Processing" tab, loading screen while processing after "Submit Request" button.

Once the input data has been processed, the API returns the outputs, which are displayed in a pop-up **results window**, as shown in Figure 64. This window consists of several elements.

First, the raw input image is shown to allow the user to clearly visualize both the original data and the processed output side by side. Second, the results from the different modules of the tool are displayed, these will only appear if the corresponding tasks were selected in the pipeline configuration.

The third element is the status vector, which contains as many values as modules were used. Each value will be 0 if the highlighted classes (for detection and segmentation) or the keywords (for VQA) were not found, and 1 if they were detected.

Finally, the results window also includes the raw JSON response returned by the API. This is particularly useful if the user wants to integrate the application with other systems or store relevant outputs, such as bounding boxes or segmentation masks, for further analysis.

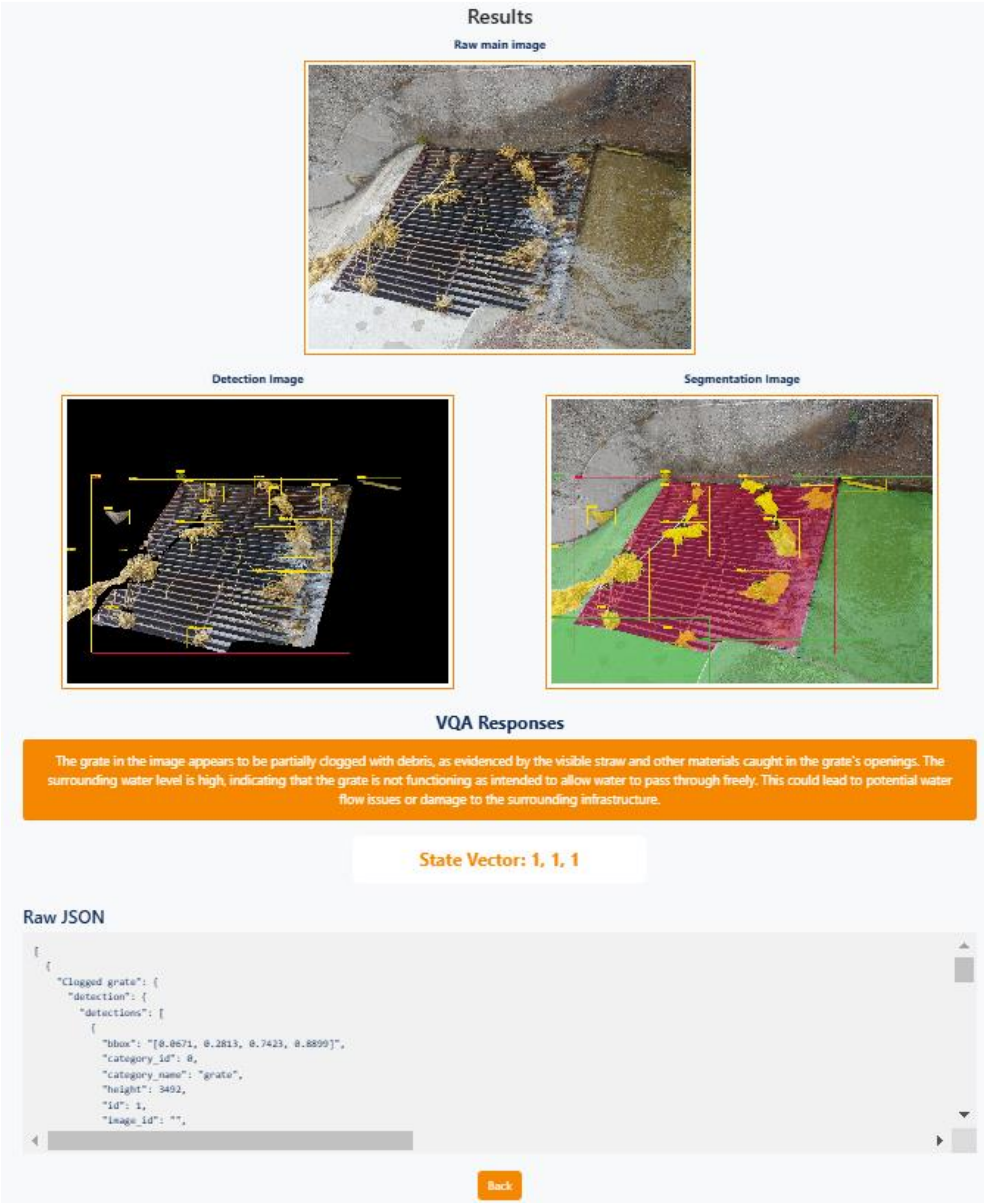

Figure 64: Results pop-up screen. Elements: Raw image, AI modules outputs, State Vector, Raw Json.

The user can interpret the results independently by analyzing the text generated by the VQA module and visualizing the processed images directly within the interface. If desired, they can open the images in full resolution in a separate browser tab, as shown in Figure 65, and download them without any restrictions.

Additionally, if the State Vector returns a value of 1,1,1, as in this case, the system will automatically send an alert to the WP7 global alert management system. This automated response occurs because the analysis suggests the presence of a "clogged grate" event, triggering the corresponding predefined alert type.

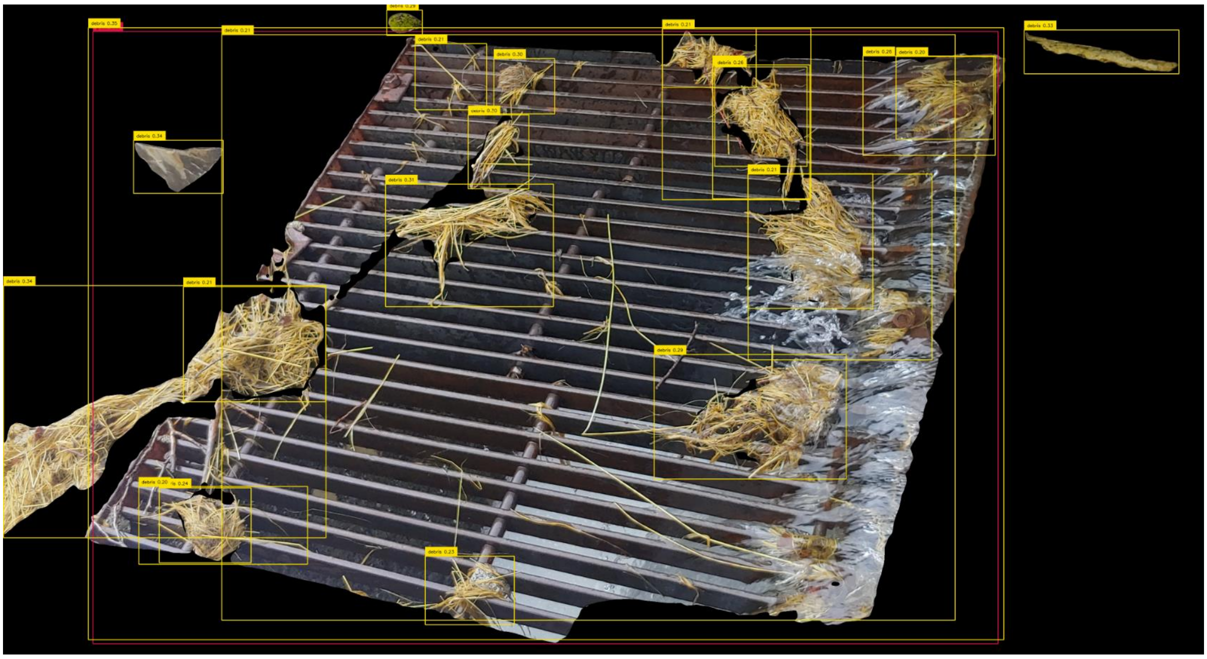

Figure 65: Results pop-up screen. Open detection image output in different window. Yellow class is "debris" and red class is "grate".

After analyzing the results, the user can return to the main menu by clicking the "Back" button in the results pop-up, Figure 64. This allows them to start a new request using the same pipeline on a different image, adjust the parameters to fine-tune the processing for the previously used image, or configure a completely different setup for another inspection use case.

Annex II: Part B – API requests and AI assistant

The UAV image processing API system provides an alternative way to interact with the UAV Remote Inspection Tool, allowing users to send requests directly to the API instead of using the graphical user interface (GUI) presented in Part A of this annex. This approach is significantly more complex, as it requires users to manually configure parameters and issue command-line requests rather than leveraging the intuitive web-based interface. While the API provides no inherent advantages over the GUI, it remains available for users who prefer or require direct control over processing workflows, particularly in automation scenarios or when integrating UAV-based inspections with other external systems.

This API system follows a modular approach, where users must first define the tasks they wish to perform—detection, segmentation, or VQA—and then specify whether these tasks should be executed independently or in a cascading fashion where the output of one module feeds into another. The API supports various pretrained AI models, allowing users to customize how images and videos are analyzed. However, the need to manually adjust parameters, select models, and construct properly formatted API requests makes this approach significantly more technical than using the GUI, which abstracts most of these complexities.

To assist users in navigating the API's complexity, a specialized ChatGPT-based assistant has been fine-tuned to generate API requests based on user input. This AI assistant understands all the API's possible parameters and can automatically generate CURL commands tailored to the user's inspection needs. Instead of manually looking up documentation and configuring every setting, users can describe their inspection scenario in plain language, and the assistant will suggest an appropriate API request. Additionally, leveraging GPT-4's multimodal capabilities, users can even upload an image of the infrastructure component they need to inspect, and the assistant will analyze the scene and provide a corresponding API request. Figure 66 shows a clear example of the AI assistant.

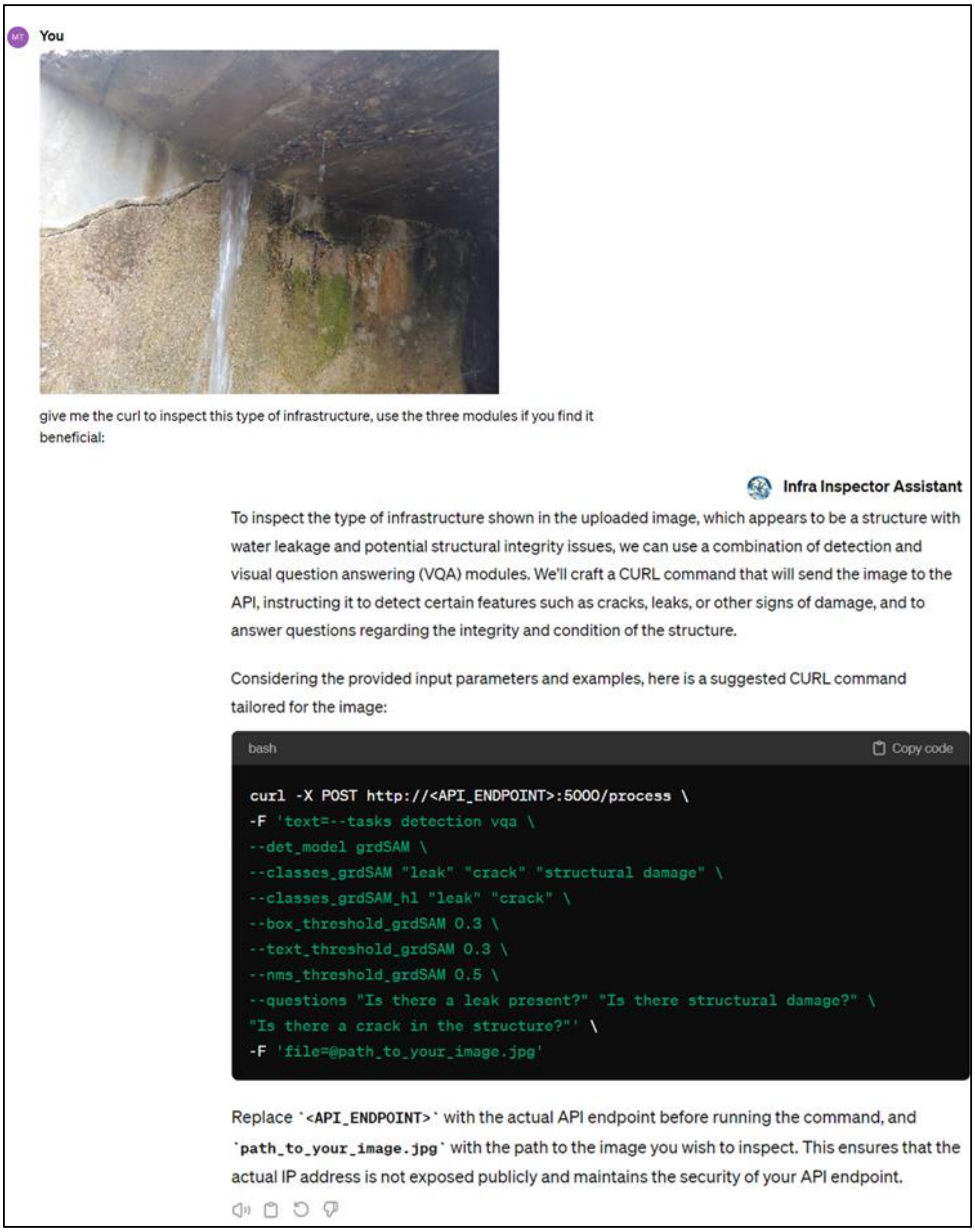

Figure 66: chatGPT Infra Inspector Assistant chatbot example. Source: [52].

Despite this AI-driven support, it is important to emphasize that using the API remains more cumbersome and less user-friendly than the GUI. The AI assistant does not replace the need for technical knowledge, and users still need to execute API calls manually. Furthermore, access to this assistant requires a ChatGPT Plus or Team subscription, meaning it is not freely available to all CI operators. While it is theoretically possible to develop an open-source alternative using state-of-the-art LLMs (Llama3.2, Phi-4, …), such an approach would require extensive dataset creation and fine-tuning, making it impractical within the scope of the current project.

To mitigate these challenges, this annex includes detailed documentation on the API's available parameters and predefined examples of commonly used API calls. Users can find ready-to-use CURL request templates for scenarios such as crack detection, fire monitoring, pipe leak detection, ceramic isolator inspection, and clogged grate assessment. These examples, combined with the virtual assistant's guidance, aim to reduce the complexity of the API's direct use, but they do not eliminate the inherent technical overhead compared to the GUI.

Given these factors, the recommended approach for most users remains the GUI-based interaction described in Part A. The API option should only be used in cases where manual API control is explicitly required, such as automating inspections through external scripts or integrating the UAV processing system into a custom workflow. For all other use cases, the GUI offers a superior user experience, providing access to the same core functionalities while removing the need for complex manual configurations.

Examples of CURLs for different inspection tasks:

- Clogged gate check:

```
curl -X POST http://api.example.com/process \
-F 'text=--tasks detection segmentation vqa \
--det_model grdSAM \
--classes_grdSAM "grate" "debris" \
--classes_grdSAM_hl "grate" \
--box_threshold_grdSAM 0.3 \
--text_threshold_grdSAM 0.3 \
--nms_threshold_grdSAM 0.5 \
--vqa_model llava \
--questions "is the grate clogged?" "is there something above the grate?" \
--seg_model_type xdecoder \
--xdec_img_size 1024 \
--vocabulary_xdec "grate" "sky" "vegetation" "soil" "people" \
--debug' \
-F 'file=@examples/clogged_grate/Clogged_grate.jpg'
```

- Fire and smoke detection:

```
curl -X POST http://api.example.com/process \
-F 'text=--tasks detection \
--det_model YOLO \
--yolo_model detection_module/models/best_fire_25000_SD.pt \
--debug' \
-F 'file=@examples/fire/fire_pexels.mp4'
```

- Custom live stream/local-cam:

```
curl -X POST http://api.example.com/process \
-F 'camera_ip=0' \
-F 'text=--tasks detection \
--det_model grdSAM \
--classes_grdSAM "people" "hammer" "glasses" "mask" \
--classes_grdSAM_hl "hammer" "sissors" \
--box_threshold_grdSAM 0.5 \
--text_threshold_grdSAM 0.5 \
--nms_threshold_grdSAM 0.5 \
--debug'
```

- Landslide detection:

```
curl -X POST http://api.example.com/process \
-F ‘text=--tasks segmentation \
--seg_model_type xdecoder \
--xdec_img_size 1024 \
--vocabulary_xdec landslide vegetation sky railway \
--debug’ \
-F ‘file=@examples/landslide/landslide_raw.jpg’
```

- Ceramic isolators inspection:

```
curl -X POST http://api.example.com/process \
-F 'text=--tasks detection vqa \
--tasks_join cascade \
--det_model grdSAM \
--classes_grdSAM "pole" "ceramic isolators" "electrical powerline"  "catenary" "tree" "plant"
"vegetation" "cable" \
--classes_grdSAM_hl "ceramic isolators" \
--box_threshold_grdSAM 0.15 \
--text_threshold_grdSAM 0.15 \
--nms_threshold_grdSAM 0.5 \
--vqa_model seal \
--questions "describe the ceramic isolators condition status" "are the ceramic isolators in good
condition" \
--vqa_triggers "broke" "rusty" \
--debug' \
-F 'file=@examples/isolators/1.jpg'
```

If YOLO is selected as the detection model, the specific models available are:

Fire and Smoke: "detection_module/models/best_fire_25000_SD.pt"

Rust/Corrosion: "detection_module/models/yolo_segx_rust.pt"

Cracks: "detection_module/models/best_50epoch_notOT_crack.pt"

If you need assistance in generating the appropriate CURL commands for your specific inspection task, you can use the dedicated virtual assistant designed for this purpose. This assistant provides guidance on parameter selection, API usage, and contextual information to help you structure your requests effectively. Please note that access requires a ChatGPT Plus or Team subscription. You can find the assistant at the following link:

https://chat.openai.com/g/g-zxGORIgXT-infra-inspector-assistant.

The following Table 12 contain a detailed breakdown of the input parameters for the main API, which orchestrates all processing tasks and ensures correct execution across different UAV-based inspection modules.

Table 12. UAV inspection tool API's input arguments.

Module	Parameter	Description	Example Usage
Main Script	**--input**	Path to the image or video file, or stream URL.	**--input 'inputs/HDE/Example.jpg'**
	--tasks	List of tasks to be performed: 'vqa', 'segmentation', 'detection'.	**--tasks vqa segmentation**
	--tasks_join	Defines the task flow: 'cascade' or 'parallel'.	**--tasks_join cascade**
	--fps	Number of frames per second to process.	**--fps 2**
	--output	Output for visualizations.	**--output 'path/to/output/folder'**
	--debug	Indicates if debug mode is activated.	**--debug True**
	--full_pipeline	Indicates if use the full pipeline approach.	**--full_pipeline True**
	--cpu	Use CPU only for processing, not GPU.	**--cpu True**
Segmentation Module	**--seg_model_type**	Type of model for segmentation: 'xdecoder' or 'grdSAM'.	**--seg_model_type xdecoder**
	--classes_grdSAM	List of classes to detect.	**--classes_grdSAM 'class1' 'class2'**
	--classes_grdSAM_hl	List of classes to highlight with background extraction.	**--classes_grdSAM_hl 'class1'**
	--box_threshold_grdSAM	Box threshold for GroundingDINO.	**--box_threshold_grdSAM 0.3**
	--text_threshold_grdSAM	Text threshold for GroundingDINO.	**--text_threshold_grdSAM 0.3**
	--nms_threshold_grdSAM	Non-maximum suppression threshold.	**--nms_threshold_grdSAM 0.5**
	--save_imgs	Option to save resulting images.	**--save_imgs True**
	--config_file_xdec	Path(s) to the config file(s) for the X-Decoder.	**--config_file_xdec 'config_file_path.yaml'**

Module	Parameter	Description	Example Usage
	--xdec_img_size	Reshape size for the image to be processed with X-Decoder.	**--xdec_img_size 512**
	--vocabulary_xdec	Concepts for segmentation with X-Decoder.	**--vocabulary_xdec 'concept1' 'concept2'**
	--vocabulary_xdec_hl	Concepts for segmentation with highlighting.	**--vocabulary_xdec_hl 'concept1' 'concept2'**
	--xdec_pretrained_pth	Path(s) to the weight file(s) for X-Decoder.	**--xdec_pretrained_pth 'path/to/weights.pt'**
	--xdec_type	Type of segmentation with X-Decoder: 'semseg' or 'refseg'.	**--xdec_type semseg**
	--pred_all_class	Option to predict all classes.	**--pred_all_class**
Detection Module	**--det_model**	Select the model used for detection: 'Detic', 'grdSAM' or 'YOLO'.	**--det_model YOLO**
	--config-file-detic	Path to the config file for Detic.	**--config-file-detic 'path/file.yaml'**
	--vocabulary	Vocabulary used in Detic: 'lvis', 'openimages', etc.	**--vocabulary custom**
	--custom_vocabulary	Custom vocabulary for Detic.	**--custom_vocabulary 'object1,object2'**
	--confidence-threshold	Minimum threshold for displaying predictions.	**--confidence-threshold 0.2**
	--nms_max_overlap	Maximum overlap threshold in NMS.	**--nms_max_overlap 0.3**
	--patching	Option to process the image in patches.	**--patching True**
	--patch_size	Size of the patches.	**--patch_size 336**
	--overlap	Overlap of the patches.	**--overlap 0.3**
	--yolo_model	Path to the YOLO model.	**--yolo_model 'detection_module/models/model_YOLO.pth'**

Module	Parameter	Description	Example Usage
VQA Module	**--vqa_model**	Select the model for visual question answering: “phi”, “llava” or “seal”.	**--vqa_model ‘phi’**
	--questions	List of questions to be answered by the AI.	**--questions 'What is in the image?'**
	--vqa_triggers	List of concepts to check if appears in the answers.	**--vqa_triggers 'problem'**
	--conv_type	Type of conversation you want.	**--conv_type ‘custom_infra’**