SUNRISE

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

D7.2 Infrastructure inspection tool and training guide V1

Document Identification			
Status	Final	**Due Date**	30/09/2023
Version	1.0	**Submission Date**	29/09/2023

Related WP	WP7	**Document Reference**	D7.2
Related Deliverable(s)	D7.1	**Dissemination Level (*)**	PU
Lead Participant	ATS	**Lead Author**	Mario Triviño, ATS
Contributors	ATS, XLAB, SKYLD, ICS, MZI, ELS, SZ, SZI, PIL, ACO, INS, HDE, TLF, TS, HER, CCL, INT	**Reviewers**	George Tsakirakis, INT

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

Document Information

List of Contributors	
Name	Partner
Mario Triviño	ATS
Dejan Štepec	XLB
Martin Pernuš	XLB
Anže Alič	XLB
Mihael Trajbarič	XLB
Yannis Arvanitakis	SKYLD
Romeo Bratska	SKYLD
Ismini Rousounidou	SKYLD
Giannis Giachos	SKYLD
Milan Tarman	ICS
Mihaela Bastar	MZI
Gorazd Azman	ELS
Tomaz Ramsak	SZ
Jakob Klofutar	SZI
Blaz Jemensek	PIL
Lars Ake Olofsson	ACO
Gilda de Marco	INS
Andrea Bello	HDE
Estanislao Fernández	TLF
Dejan Soster	TS
Christine Nam	HER
Laura Daly	CCL
George Tsakirakis	INT

Document History			
Version	Date	Change editors	Changes
0.1	11/07/2023	Mario Triviño (ATS)	Table of Content included, and first round of general document general data.
0.2	02/08/2023	Mario Triviño (ATS)	First round of content in Sections: 1 (1.1, 1.2, 1.3), 3(3.1, 3.2, 3.3, 3.5, 3.6)
0.3	21/08/2023	Martin Pernuš (XLB), Dejan Štepec (XLB), Anže Alič (XLB)	Section 2 content added.
0.4	22/08/2023	Mario Triviño (ATS)	Second round of content in Sections: 1 (1.1, 1.2, 1.3), 3(3.1, 3.2, 3.3, 3.4, 3.5, 3.6), 5
0.5	28/08/2023	Romeo Bratska (SKYLD)	Input of content in section 3(3.6.1, 3.6.2, 3.6.3,3.6.4), 4(4.1,4.2,4.2.1,4.2.2,4.3, 4.4, 4.5)
0.6	07/09/2023	Mario Triviño (ATS)	Final round of content in all sections and first full draft integration

0.7	08/09/2023	Giannis Giachos (SKYLD)	Final Formation of content in section 3(3.6.1, 3.6.2, 3.6.3,3.6.4), 4(4.1,4.2,4.2.1,4.2.2,4.3, 4.4, 4.5)
0.8	08/09/2023	Mario Triviño (ATS)	Final integration and review
0.9	20/09/2023	Mario Triviño (ATS)	Changes made after review, format and added references
0.10	28/09/2023	Mario Triviño (ATS)	Version candidate for submission
0.11	29/09/2023	Antonio Álvarez (ATS)	Final document layout
1.0	29/09/2023	Antonio Álvarez (ATS)	FINAL VERSION TO BE SUBMITTED

Quality Control		
Role	Who (Partner short name)	Approval Date
Deliverable leader	Mario Triviño (ATS)	28/09/2023
Quality manager	Juan Andrés Alonso (ATS)	29/09/2023
Project Coordinator	Antonio Álvarez (ATS)	29/09/2023

Table of Contents

List of Tables

List of Figures

List of Acronyms

Abbreviation / acronym	Description
3D	Three Dimensional
AES	Advanced Encryption Standard
AMSL	Above Mean Sea Level
AOI	Area of Interest
API	Application Programming Interface
BC	Binary Change
BiT	Bitemporal image Transformer
CAN	Controller Area Network
CDLR	Change detection based on image Reconstruction Loss
CI	Critical Infrastructure
CNN	Convolutional Neural Network
COCO	Common Objects in COntext
D7.1	Deliverable number 1 belonging to WP 7
DMIC	Digital Microphone
DSPK	Digital Speaker
EC	European Commission
eMMC	Embedded Multimedia Card
EO	Electro-Optical
ESA	European Space Agency
GbE	Gigabit Ethernet
GCS	Ground Control System
GLONASS	Global Navigation Satellite System
GNSS	Global Navigation Satellite Systems
GPIO	General Purpose Input/Output
GPS	Global Positioning System
GUI	Graphical User Interface
IR	Infra-Red
IoU	Intersection over Union
LiDAR	Light Detection and Ranging
LLM	Large Language Models
LOS	Line of Sight
LRF	Laser Range Finder
MAE	Mean Absolute Error
mAP	Mean Average Precision
mIoU	Mean Intersection over Union
MQTT	Message Queuing Telemetry Transport
MSE	Mean Squared Error
NIR	Near-InfraRed

POI	Point Of Interest
PWA	Progressive Web Applications
PWM	Pulse Width Modulation
QoS	Quality of Service
REST	Representational state transfer
RF	Radio Frequency
RGB	Red Green Blue
SAR	Synthetic Aperture Radar
SC	Semantic Change
SCS	Semantic Change Segmentation
SDG	Synthetic Data Generation
SNR	Signal-to-Noise Ratio
SOTA	State-Of-The-Art
SPA	Single-Page Applications
STS	Security Token Service
TRL	Technology Readiness Level
U-TAE	U-net with Temporal Attention Encoder
UAV	Unmanned Aerial Vehicle
UI	User Interface
VHM	Vegetation Height Model
VLM	Visual Language Models
ViT	Visual Transformer
VQA	Visual Question Answering
WP	Work Package

Executive Summary

This document introduces the initial iteration in the process of implementing remote inspection tools for critical infrastructures. Within the scope of WP7, it represents the inaugural effort to implement the concepts introduced in D7.1, aiming to address all the requirements of CI stakeholders outlined in D3.2. The progression of the content herein will be evident in subsequent iterations (D7.3-D7.6).

This is a technical report wherein each primary section is dedicated to detailing a component of the comprehensive remote inspection module, namely: the satellite imaging inspection module, the UAV imaging inspection module, and the user interface that facilitates interaction with the aforementioned modules and displays the results.

The proposed solutions leverage artificial intelligence computer vision neural models for image analysis to extract data regarding the infrastructure's condition. The content provides an examination, execution, and testing of state-of-the-art models in critical tasks such as vegetation monitoring, anomaly detection, open-vocabulary object detection/segmentation, Visual Question Answering (VQA), and 3D reconstruction. Among the models incorporated in the suggested solutions are some of the most groundbreaking developments from leading industry firms, Visual Language Models (VLM) and Large Language Models (LLM), featuring cutting edge architectures like Transformers. Thus, this document allows readers to discern the applicability and adaptability of these significant advancements to specific challenges in real-world scenarios.

The development status of the SW solutions aligns with expectations, with the PoCs of "Pilot 0 - lab validation" achieving a TRL of 5 or higher, and the laboratory integration of the UAV platform successfully completed, encompassing all essential inspection components. The outcomes from the two visual image inspection sub-modules indicate that the current tool can address, wholly or partially, challenges in real-world settings. Furthermore, the delineation of the module architectures and the deployment plan that are included, substantiate a well-defined roadmap towards achieving the set objective.

With all the content provided, D7.2 establishes a robust foundation upon which further development can ensue, incorporating additional features and models to enhance usability and address as many inspection challenges as possible.

1 Introduction

1.1 Purpose of the document

The SUNRISE project is designed to bolster the resilience of essential services within Europe's Critical Infrastructure (CI), equipping operators and authorities with the necessary tools to handle situations similar to those encountered during the recent pandemic. To achieve this, the project proposes the development of a strategy and a series of tools, all aimed at ensuring service resilience and the continuity of operations. Included in this set of tools is the inspection module of WP7, which primarily focuses on the implementation of solutions for remote inspection of critical infrastructures.

In this context, the main goal of D7.2, *"Infrastructure inspection tool and training guide V1",* is to clearly and succinctly outline the approach and the initial steps taken during the development of the tools designed to inspect key elements and structures using information collected remotely via satellites or UAVs. The description of the document included in the DoA states "Initial infrastructure inspection tool's components tested in labs and released with usage guidelines" which is the purpose and the content included.

Based on this, D7.2 aims to serve as a foundation for introducing and justifying the proposed solutions during the first-year review of the SUNRISE project. It also acts as a support document, guiding the user in the application of the proposed solutions. This document, technically focused, also intended to serve as a reference guide if there is a need for detailed information about the methods, architectures, integrations, hardware components, and algorithms implemented in this initial project phase.

Therefore, this text seeks to describe the current state of the visual inspection module at M12 of the project. This includes the initial conceptual approach, the current development status of the submodules, and the results obtained during the project's Pilot 0. The latter includes laboratory tests using open-source data, available datasets, and small-scale proof-of-concept tests on actual data manually collected at the pilot facilities. By M12 month of the project, the technology presented in this deliverable should reach a Technology Readiness Level (TRL) of 5, indicating that the technology has been verified in a relevant environment.

1.2 Relation to other project work

This deliverable represents the second output envisaged from WP7, and thus, reflects the work related to the tasks it encompasses, namely: T7.1 Visual infrastructure inspection with satellite images; T7.2 Visual infrastructure inspection with UAVs; T7.3 User interfaces for remote infrastructure inspection; T7.4 Continuous integration and testing; and lastly, T7.5 Demonstration, training, evaluation, and validation. WP7, like all technical WPs, is interconnected with the management WP (WP9), dissemination and exploitation (WP8), ethical requirements (WP10), collaboration (WP1), strategy (WP2), and design (WP3). Close and necessary collaboration with all these work packages is in place, but it's important to emphasize the primary link with WP3, which is responsible for gathering requirements, maintaining unity, and serving as an interface between the different modules of the overall tool designed in each of the technical WPs.

Concerning already delivered documents and those currently in progress, D7.2 has a strong relationship with D7.1 and D3.2. On one hand, D7.1, "Infrastructure inspection conceptualization," is the foundation upon which the developments of this document are based, as it provides a detailed introduction to the pilots and their various challenges (D7.1 - chapter 2), and presents the conceptual plan proposed for the development of the tools of this module (D7.1 - chapter 3). On the other hand, D3.2 incorporates the second iteration of the requirements extracted during the needs analysis of the CI stakeholders and the technical approach of the solutions, and as such, it represents a significant input for the tools described in this deliverable.

1.3 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. The other five include:

- **Chapter 2** provides the general context and a high-level design of the satellite inspection tool's architecture. It describes the tool's modules, namely 'Infrastructure change detection' and 'Vegetation monitoring', presents their lab validation, and discusses deployment strategies.
- **Chapter 3** introduces the context, high-level design, and description of the UAV inspection tool. It contains a detailed breakdown of the tool's modules, including 'Object detection and semantic segmentation', 'VQA' (Visual Question Answering), and '3D virtualization', along with their respective lab validation and deployment details. It concludes with a discussion on UAV platform lab integration.
- **Chapter 4** discusses the general context, high-level design of the user interface architecture for remote infrastructure inspection and presents mockups of the dashboards. It details the process of integration and validation, as well as the deployment strategy.
- **Chapter 5** covers the execution of lab pilot trials, updating the content introduced in D7.1.
- **Chapter 6**, final chapter, offers an overall summation of 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, they are all part of the same remote inspection module, so the integration of the tools at the end of the project must be complete.

2 Satellite inspection tool

This module will provide AI-supported satellite-imagery-based CI inspection tools. Satellite imagery and accompanying AI tools provide the ability to monitor the physical infrastructure in a non-invasive and continuous manner. This can also serve as a trigger to activate the more costly and labour-intensive UAV-based inspection (Section 3), which is offering a more targeted, localized area-focused monitoring of regions of interest. This section builds upon D7.1 by introducing individual sub-modules in more detail (Sections 2.3.1 and 2.3.2), as well their lab validation results at TRL-5 level (Section 2.4).

2.1 General context

The satellite-imagery-based inspection module uses satellite imagery of different modalities to enable large-scale, non-invasive continuous inspection of the physical infrastructure. The main innovation of the satellite-based inspection module is presented by i) the use of satellite imagery to optimize manual physical inspections of the CI and ii) the usage of AI to automate the processing of the satellite imagery data itself. The automated processing of the satellite imagery is achieved by using computer vision to process satellite images to detect different events and monitor changes. We are focusing on remote infrastructure inspection by means of different satellite imagery modalities (optical, multispectral) to enable different application use-cases. The modalities and sensors were selected based on the initial discussions with the pilots and their identified problems that require remote inspection, presented in D7.1.

The identified cross-pilot problems identified in D7.1 can be separated into i) vegetation management and ii) infrastructure change monitoring. Vegetation monitoring requires specific solutions which can identify the vegetation and quantify its threat to the infrastructure. We provide a solution that can detect vegetation and estimate its height from readily available multispectral satellite-imagery data. The detected vegetation and its height can be directly used for threat estimation, according to the vicinity of the CI. In D7.1 we also identified many problems that relate to the changes around the CI (e.g., landslides, clogging, leaks, illegal build-ups). We provide a general solution of infrastructure change monitoring which is not task-specific and enables a general detection of visual change around CI using satellite imagery data.

2.2 Architecture: high level design

As already introduced in Section 2.1, this module consists out of i) infrastructure change detection and ii) vegetation monitoring sub-modules. The sub-modules are independent and connected to the main satellite-based inspection module, which handles data collection and pre/post-processing, as well as interaction with other components (e.g., dashboards and pilot integrations in T7.3 and UAV-based inspection in T7.2). Figure 1 shows this architecture visually.

Pre-processing consists of data collection from various satellite imagery providers and conversion of the collected data into format(s) that is applicable.

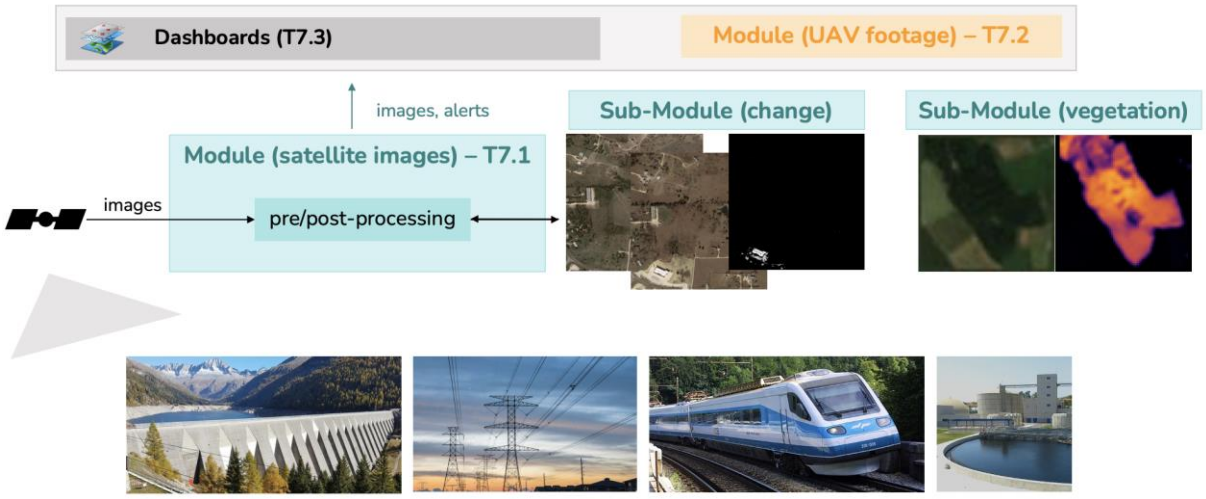

Figure 1: High-level architecture of the satellite-based inspection module and its sub-modules (T7.1).

In the following sections we describe individual infrastructure change detection (Section 2.3.1) and vegetation monitoring (Section 2.3.2) sub-modules in detail.

2.3 Tool modules description

2.3.1 Infrastructure change detection

Regular inspections of critical infrastructure are essential to ensure their integrity and functionality. Manual inspections have drawbacks such as being time-consuming, expensive, and infrequent. They are usually time-based instead of risk-based, which reduces the effectiveness of such inspections. Furthermore, these limitations can lead to delayed detection of significant damage, subsequently escalating the risk of severe harm to vital infrastructure systems. As such, there is a need for automatic methods to proactively detect potential risks and enhance the overall resilience of critical infrastructure.

Using the power of advanced satellite technology, it becomes possible to monitor infrastructure sites from a distance, facilitating more frequent and efficient assessments, which can be performed on-demand or continuously, depending on the nature of the monitored infrastructure, possible threats, and cost effectiveness. This not only helps overcome the limitations of manual inspections but also significantly reduces the risk of overlooking critical issues. With machine-learning and image processing, by comparing two satellite images taken at different points in time, changes in critical infrastructure can be detected (e.g., earth movements, landslides, debris coming from upstream, vegetation overgrowth). Due to the dynamic environment in which critical infrastructure is built, the main challenge is to detect any kind of undesirable change.

2.3.1.1 Change detection data

Datasets for satellite imagery change detection must be large enough so that we are able to train and evaluate machine learning models. Ground sample distance should be sufficiently small (e.g., <= 10m) so that we are able to detect smaller changes near critical infrastructure. Three open-source datasets dedicated to satellite imagery change detection were identified, offering significant potential for training, and evaluating machine-learning algorithms suitable for our use case and for comparing the developed methods against state-of-the-art.

The first dataset, **DynamicEarthNet** [14] (Figure 2), presents a comprehensive and unique resource for satellite imagery analysis. It comprises a collection of daily, cloud-free satellite images captured between January 2018 and December 2019; the dataset encompasses 75 diverse areas of interest spanning the globe. Each area of interest is represented by a sequence of 730 images, resulting in a

total of 54,750 satellite images. The images are obtained from the Fusion Monitoring product by Planet Labs [53] and include four spectral channels (RGB + NIR) with a pixel resolution of 3 meters. Notably, the dataset incorporates pixel-wise semantic labels that define land cover changes. These labels are available for the first day of each month and include predefined categories such as impervious surfaces, agriculture, forest & other vegetation, wetlands, soil, water, and snow & ice. With its detailed annotations and global coverage, DynamicEarthNet serves as a valuable resource for research in satellite data analysis, enabling the exploration of both short-term and long-term land cover changes.

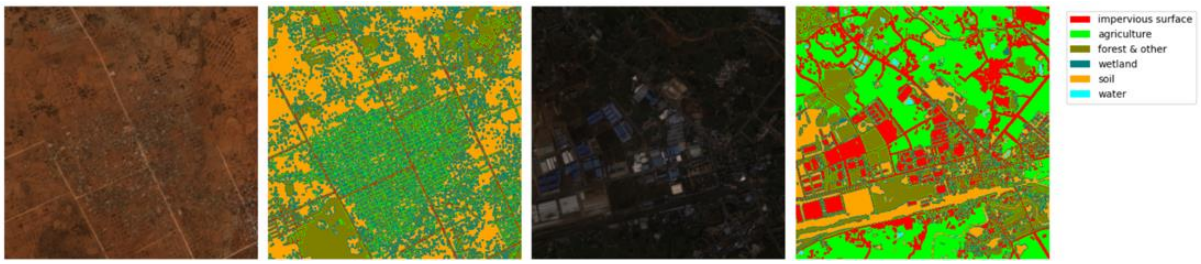

Figure 2: Two examples extracted from the DynamcEarthNet dataset [14], each paired with its corresponding semantic segmentation mask.

The second dataset is the **xBD** dataset [15] (Figure 3), a large-scale dataset for change detection and building damage assessment. xBD offers pre- and post-event satellite imagery across a variety of disaster events with building polygons, ordinal labels of damage level, and corresponding satellite metadata. It includes satellite imagery of earthquake, tsunami, flood, volcano, wildfire and tornado/hurricane disaster events across sixteen regions. Furthermore, the dataset contains bounding boxes and labels for environmental factors such as fire, water, and smoke. It contains 850,736 building annotations across 45,362 km^2 of imagery.

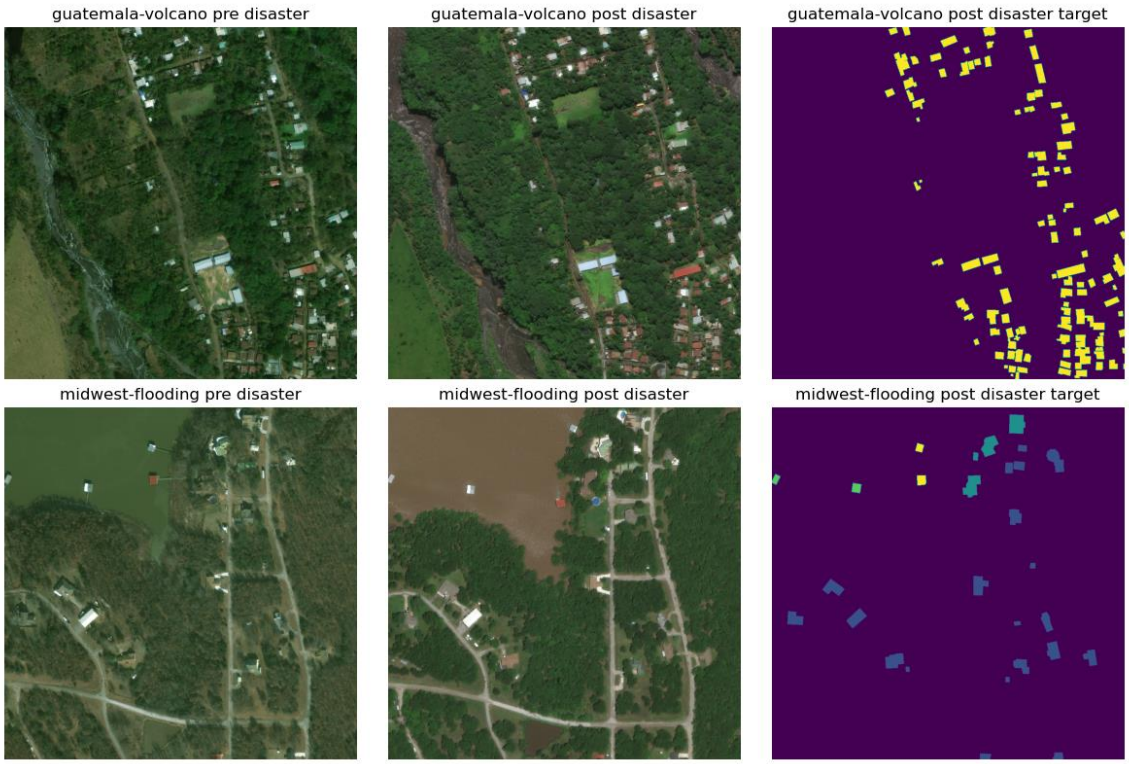

Figure 3: Two examples from xBD dataset [15]. The image on the left shows the initial state (before), while the middle image shows the subsequent state (after). The image on the right displays the ground truth data for building damage assessment.

The final dataset is the **LEVIR-CD** dataset [16] (Figure 4). It consists of 637 pairs of high-resolution Google Earth image patches, each measuring 1024 × 1024 pixels. These images depict temporal changes spanning 5 to 14 years, with a specific focus on significant building growth. Covering various building types, the dataset's annotations emphasize changes in building status, both growth and decline, marked by binary labels (1 for change, 0 for unchanged). The dataset encompasses a total of 31,333 instances of individual building changes.

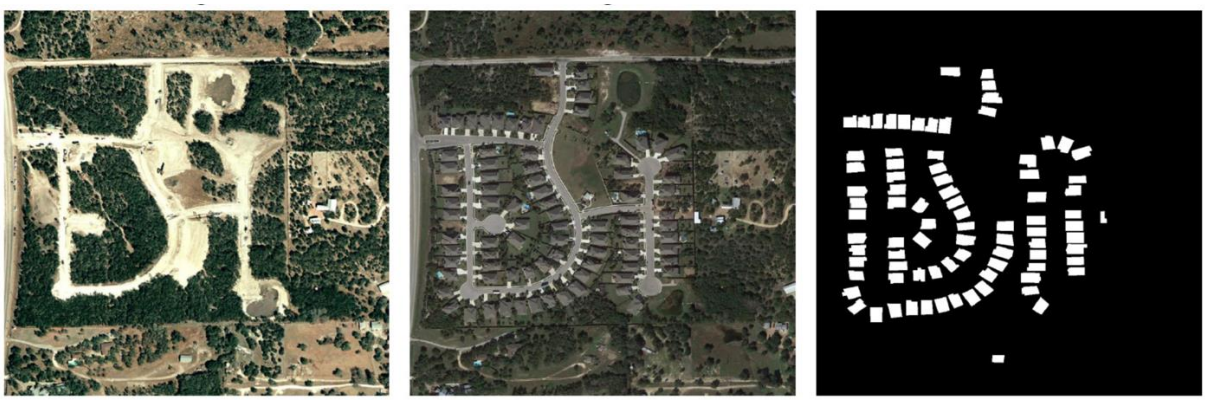

Figure 4: Example from the LEVIR-CD dataset [16], illustrating instances of building change detection. The image on the left shows the initial state (before), while the middle image shows the subsequent state (after). The image on the right displays the ground truth, denoting the spatial information about the region changes.

While the described change detection datasets have played an instrumental role in the progression of this field, providing invaluable data that enabled the development of new methodologies and approaches, they also have their limitations. One of the most significant limitations is their reliance on a predetermined set of categories. This design inherently constrains the range of changes that can be detected, potentially overlooking novel or unexpected shifts in the environment.

Our aim is to develop techniques that can operate not only on known categories but also on previously uncharted ones. This can be achieved by developing methods that can detect any kind of terrain or infrastructure change. Therefore, the aforementioned datasets will be used only to quantify the performance of our approaches. If the existing datasets will be deemed as unsuitable due to their limited change annotations, we will create a new dataset that will serve as a foundation for the methodology evaluation. Additionally, by receiving data from pilot partners on historical events that led to landscape and infrastructure alterations, we can adjust the evaluation to be more relevant for specific use cases.

2.3.1.2 Methods overview

Satellite image change detection involves two primary methodologies: **supervised** and **unsupervised** methods. In the supervised paradigm, machine learning algorithms learn from meticulously labelled training data, where each image pair is annotated to highlight both changed and unchanged areas. It is crucial to emphasize that supervised techniques necessitate the annotation of every pixel in the images under study. This labour-intensive approach demands human annotators to categorize each pixel based on the perceived change between the satellite images. With these annotations as a reference, the algorithm identifies recurring patterns in new image pairs to pinpoint changes. Although supervised methods can achieve highly accurate results, they are confined to detecting changes within the categories predefined during training.

On the other hand, **unsupervised** methods sidestep the need for labour-intensive labelled data. These methods analyse statistical deviations between images to flag discrepancies, detecting changes across a wide spectrum—regardless of whether they were encountered during training. The flexibility of this

approach enables it to adapt to novel change types, although this versatility might lead to some precision trade-offs, including potential false positives or overlooking subtle changes.

An important consideration here is the correlation with the unsupervised visual anomaly detection approach, which boasts an established protocol and dataset (e.g., MvTec [17]). This diverges from the realm of remote sensing, where a concrete unsupervised protocol is notably absent, despite the broad utility, particularly within real-world contexts for such methodologies. The primary distinction lies in the fact that anomaly detection operates on individual images, whereas change detection leverages paired images. Nonetheless, the methodologies from anomaly detection can be suitably extended to serve to our specific requirements.

2.3.1.3 Methods for change detection

Change detection methods in computer vision are designed to identify changes between two or more images. A machine learning model typically accepts two or more images as input and predicts if change happened or not for each pixel. Methods for change detection are inspired by image segmentation methods. However, whereas image segmentation focuses solely on an individual image, change detection also considers the temporal dimension. We evaluated the following methods which take as an input two images (bi-temporal setting):

- **ChangerEx** [18] focuses on the "exchange" of bi-temporal features, emphasizing mutual learning through feature exchange and mixing layers. This approach promotes automatic domain adaptation between two temporal domains instead of strictly adhering to a temporal sequence for change detection.
- **BiT** (Bitemporal image Transformer) [19] condenses intricate image changes into a handful of significant visual concepts, termed tokens. These tokens are then processed using the Transformer architecture [11].

And models that take as an input three images or more:

- **3-D U-Net** [20] builds upon the foundational U-Net model [4], a popular convolutional neural network design in computer vision known for its semantic segmentation capabilities. Characterized by its U-shaped architecture, the U-Net model features a contrasting path, a central bottleneck, and an expansive path. The 3D U-net applies 3D convolution to process data over time dimension, enabling the network to handle spatiotemporal information.
- **U-Net with Temporal Attention Encoder (U-TAE)** [21] model encodes image sequences through a shared spatial convolutional encoder and incorporates a temporal attention encoder to generate attention masks which captures essential features. By integrating information from all images, the attention-based fusion enables the model to predict changes effectively.

The field of unsupervised methods for remote sensing is an emerging research domain, where the following method has been introduced:

- **CDLR** (Change Detection based on image Loss Reconstruction) [22] method utilizes image reconstruction, operating with only a single-temporal unlabelled image. The model is trained to reconstruct the original image from the input image and a generated augmented image. During inference, it identifies the changed regions between bi-temporal inputs by noting regions with high image reconstruction loss.

2.3.2 Vegetation monitoring

2.3.2.1 Satellite Providers

In the development of our tool for satellite-based vegetation monitoring, there are several satellite-imagery providers to choose from. The decision regarding the providers was guided by the following considerations:

- **Cost-effectiveness.** The satellite provider should offer data without excessive expenditure.
- **Historical archives**. Historical satellite imagery is essential for developing a machine-learning solution, as these algorithms require a substantial volume of data to train efficiently.
- **Resolution.** By offering imagery at a higher spatial resolution, the finer details of vegetation structure can be captured and analyzed.

Considering these goals, we leveraged the capabilities of two state-of-the-art satellite providers/constellations: **ESA Sentinel-2** and **Planet PlanetScope**.

Sentinel-2: Sentinel-2 is a component of the European Space Agency's (ESA) Copernicus Programme[1], a project that provides global high-quality satellite imagery. These images support service providers, governmental entities, and various organizations. Copernicus primarily focuses on areas such as the atmosphere, marine ecosystems, land, climate, emergency management, and security. It consists of two satellites and offers high-resolution observations with spatial resolutions ranging from 10 to 60 meters, and revisit time of 2-3 days in mid-geographic regions. The thirteen satellite spectral bands encompass visible, near-infrared (NIR) and shortwave infrared spectrums.

Planet PlanetScope: PlanetScope is operated by a private company Planet and comprises of a constellation of small satellites. The satellite boasts a rapid revisit time to any given location. The imagery has resolution of 3-5 meters, allowing for detailed observations of Earth's surface. Until 2021, the imagery was provided in four spectral bands. In 2021, Planet upgraded the original satellites, adding four additional spectral bands.

Owing to the availability of high-resolution Sentinel satellite imagery data, and its unrestricted access, a substantial amount of research has been conducted using this information [23][24][25][26][27]. Among these studies is a work on Sentinel-based vegetation height prediction [3], providing an opportunity for a direct comparison of our proposed methodologies with existing state-of-the-art techniques. Consequently, we report results exclusively on Sentinel imagery.

2.3.2.2 Vegetation Height Data

To develop a machine-learning solution, we require ground truth data that allows us to match a given satellite image to a corresponding vegetation height map. For this purpose, we use **Vegetation Height Model (VHM)**[2] by National Forest Inventory [1], which was calculated for entire Switzerland using digital aerial images. Due to the similarity of vegetation characteristics between Swiss and target areas, the trained machine-learning models are expected to generalize well. VHM data contains a very high spatial resolution of 1x1 meters. The measurements were taken during the summer period over six years. Besides raw measurements, VHM data includes metadata about the location and the time of every measurement.

2.3.2.3 Data collection

We developed a meticulous data processing pipeline to match each VHM data point to its corresponding satellite imagery that was obtained from the satellite imagery provider's archive. Only imagery with less than 5% cloudiness was considered to ensure the reliability of the data. Acknowledging the temporal dynamics of vegetation growth, a two-week window was defined

[1] All Copernicus Sentinel data is released under Creative Commons CC BY-SA 3.0 IGO licence.
[2] Released under Open Database Licence (ODbL) licence.

between the target data point and the available satellite imagery. This served as a guideline to identify suitable images.

The satellite imagery was searched based on the time-window and the closest image was selected from available options to ensure the most accurate representation of the vegetation at that point in time. If no suitable images were identified, the data point was discarded. This carefully designed pipeline ensures that the dataset maintains a high level of precision and relevance.

Sentinel-2 satellite imagery was further processed with ESA's toolbox *sen2cor* [2], which performs atmospheric effect correction. By minimizing inconsistencies and distortions that may arise from factors like atmospheric haze, moisture, or other factors, the processed images provide a more faithful representation of the actual vegetation structure and decrease imagery variability, which can contribute towards performance of vegetation height prediction solution.

The Sentinel-2 imagery consists of satellite channels with varying resolution. The highest available resolution is 10 meters, but some channels contain data with 20-meter or 60-meter resolution. Such lower-resolution data was resampled to 10 meters using bilinear interpolation. On the other hand, all channels of Planet imagery have a resolution of 3-5 meters.

Each obtained satellite imagery was paired with the corresponding satellite mask. The satellite mask is additional satellite data detailing the spatial information about the observed area, such as the presence of clouds, water bodies, or snow. These masks are employed to direct the machine learning method towards specific areas of interest, meticulously filtering out irrelevant or unwanted regions from the final data, enhancing the precision of the developed method.

Some examples of extracted Sentinel-2 satellite imagery and the matching vegetation height map are shown in Figure 5.

2.3.2.4 Data processing

The dataset was divided into training, validation, and test splits according to the established procedure. Concretely, the test split was defined according to the test area as described in [3]. The rest of the area was defined as a training and validation area. The validation samples are drawn randomly from the non-test area. The final data split is shown in Figure 6. The training set was used to calculate various statistics of the dataset to normalize the data during machine learning method training.

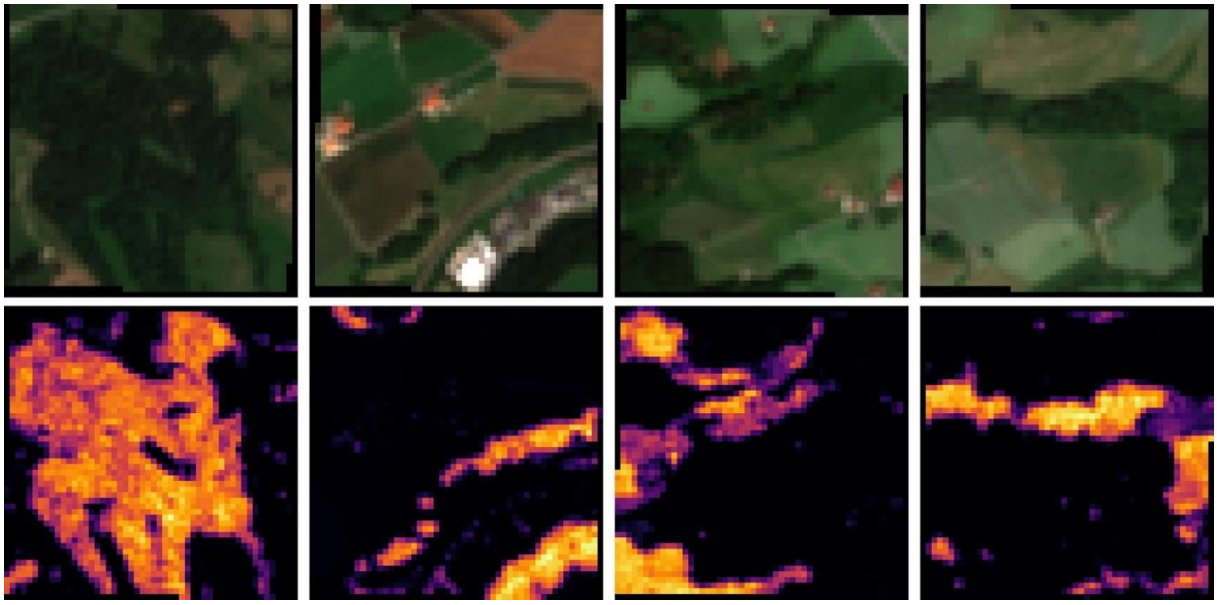

Figure 5: Examples of Sentinel-2 satellite imagery [54] (top) and the corresponding Vegetation Height Model data [1] (bottom).

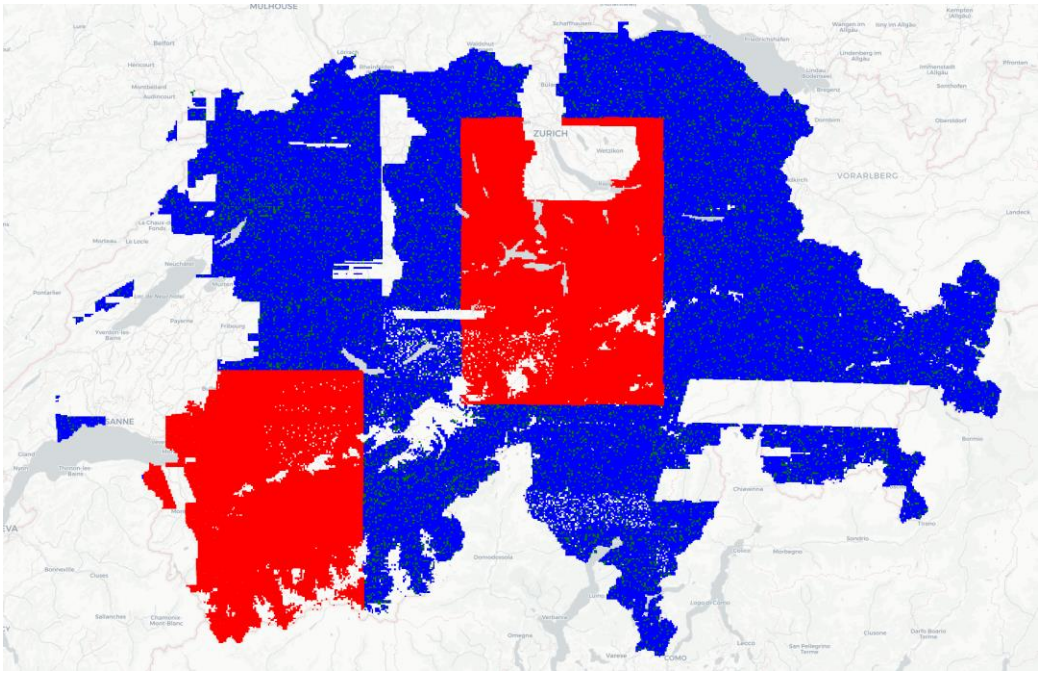

Figure 6: Data split of the collected dataset. Blue, green, and red colour denote the training, validation, and test split, respectively.

In addition to utilizing the raw satellite measurements, our approach incorporated a series of data transformations, tailored towards various data bands. These transformations define mathematical operations on specific data bands with the aim of producing a new data band that provides more relevant data towards the desired task. For example, a transformation on near-infrared band and red light defines a Normalized Difference Vegetation Index, which highlights the vegetation in a certain area. We included several such transformations to provide better data representation for our machine-learning methods, thereby enhancing the feature space that our models can leverage.

2.3.2.5 Machine-learning methods

The task of vegetation height prediction is closely related to the popular computer vision task of semantic segmentation. Both tasks involve making pixel-wise predictions, where the main difference lies in whether the model prediction is of discrete nature, as is the case in semantic segmentation task, or a continuous one, as in the case of vegetation height prediction. Therefore, we repurposed such models towards our task.

One notable characteristic of such models is their division into two main components: the backbone and the head. The backbone is primarily responsible for feature extraction. It interprets an input image and transforms it into a set of features that can be utilized for various tasks. Given its crucial role, the choice of backbone can significantly influence the model's performance. The head, on the other hand, makes use of these features to perform the actual task, such as vegetation height prediction as in our case.

In our experimentation, we explored various architectures, both in terms of their overall structure and the specific backbones they employed. Specifically, some of the most prominent architectures and backbones we tested are:

- **UNet** [4] is a fully convolutional network that was originally purposed for the task of biomedical image segmentation. Its architecture is characterized by a symmetrical contracting path that captures context and a symmetric expanding path that enables precise image predictions. We employ EfficientNet [13] as the backbone model.

- **DeepLabv3** [7] is an enhanced fully connected network that employs atrous convolution [55] with various dilation rates, which allows it to capture multi-scale information.
- **Swin Transformer** [6] applies shifted windows to partition images into non-overlapping local regions, which are processed with Transformer architecture [11]. Swin Transformer is used as a backbone and UPerNet [10] is used as the head model, as proposed in the original paper.
- **ConvNeXt** [5] is a modernized convolutional architecture that introduced several key model components, which achieved superb classification, semantic segmentation, and object detection performance. ConvNeXt is used as a backbone model and UPerNet [10] is used as the head model, as proposed in the original paper.

We chose the Mean Absolute Error (MAE) as our principal training loss and evaluation benchmark. When compared to the frequently adopted Mean Squared Error (MSE) criterion, MAE presents distinct advantages. Specifically, within the VHM dataset, which contains occasional outliers due to measurement inaccuracies, MAE proves to be less susceptible. Additionally, while MSE can often lead to smoother or blurrier predictions, MAE reduces the likelihood of such over-generalizations, yielding a more faithful portrayal of vegetation height variations.

2.4 Tool modules lab validation

In this section we evaluate the developed approaches against state-of-the-art solutions in the literature. This has enabled us to select appropriate approaches, as well as to test our newly developed solutions in a reproducible and comparable fashion. Note that real-world pilot data for the evaluation purposes will be scarce (e.g., events of interest are rare), making it hard to comprehensively evaluate the performance of the developed approaches at SUNRISE pilot sites.

2.4.1 Infrastructure change detection

In this segment, we begin by introducing the evaluation metrics utilized in the domain of change detection. We then proceed to evaluate the identified methodologies and assess their suitability within the context of our unique use case.

2.4.1.1 Evaluation metrics

In the context of change detection, precision, recall, and the F1 score [57] play a crucial role in evaluating the effectiveness of the machine-learning model. Precision assesses the accuracy of detected changes, indicating how many of the reported changes are indeed valid. Recall measures the ability to detect all actual changes, ensuring that important changes aren't missed. The F1 score provides an overall assessment of the model's ability, as a balance between precision and recall, to identify changes accurately and comprehensively in the given data.

Another commonly used metric is Intersection over Union (IoU) [56], which assesses how accurately predicted changed areas align with actual changes. It quantifies the overlap by calculating the ratio of the intersection to the union of the regions, comparing ground truth and predicted change. This evaluation method measures the effectiveness of change detection algorithms, with IoU values ranging from 0 to 1 where score of 1 signifies a flawless prediction. When dealing with multiple categories, a commonly adopted approach is to utilize the *mean Intersection over Union* (mIoU) metric. This metric computes the average of the IoU values for each category, providing a consolidated assessment of change detection accuracy across various classes.

When pixels are also annotated with a class label, the Semantic Change Segmentation (SCS) metric can be used. It comprises two key aspects in evaluating change detection results: binary change (BC) and semantic change (SC). BC quantifies the alignment of predicted change with actual change, utilizing the IoU to measure overlap. SC assesses semantic accuracy among changed pixels, calculating the Jaccard index for predicted labels compared to ground-truth labels within the set of changed pixels.

The SCS score is a mean of BC and SC and offers a comprehensive measure of the quality of semantic change segmentation.

It is important to highlight that in this deliverable, our focus revolves around the comparison of state-of-the-art change detection methods. Nevertheless, our future work will predominantly focus on event detection that has already occurred. During this phase, we intend to adopt more user-centric evaluation protocols, which may be less rigorous, to evaluate practicality of the system from an end-user perspective.

2.4.1.2 Experiments

Method evaluated that work with bi-temporal setting were evaluated on LAVIR-CD dataset. Results can be seen in Table 1.

Table 1: Quantitative results of change detection on LEVIR-CD dataset [16].

Method	IoU (%)	Precision (%)	Recall (%)	F1 (%)
BiT	80.86	89.24	89.37	89.31
ChangerEx	85.76	92.97	90.61	91.77
CDLR	59.0	63.0	92.0	74.78

Due to the LAVIR-CD dataset's exclusive focus on building changes, the CDLR method achieves lower F1 and IoU scores. However, CLDR also identifies additional changes not labeled in the dataset, underscoring its broader applicability. A visual example of CDLR method performance in shown in Figure 7. This aspect is evident in its higher recall compared to other supervised methods, showcasing the promise of unsupervised approaches in change detection. In the realm of supervised methods, ChangerEx outperforms BiT and currently holds the position of the state-of-the-art (SOTA) for change detection.

Figure 7: Example from the LEVIR-CD dataset [16], illustrating detected changes using the CDLR method [22]. The image on the left shows the initial area state, while the middle image shows the later state with newly constructed buildings present in the image. The image on the right displays the predicted changes with CDLR method.

We evaluated 3D-Unet and U-TAE on DynamicEarthNet dataset which provides a model as an input multiple images so that the model has an additional context with images on daily or weekly basis. The results can be seen in Table 2. Examples of predictions can be seen in Figure 8.

Table 2: Quantitative results of semantic change segmentation on DynamicEarthNet dataset [14].

Method	SCS (%)	BC (%)	SC (%)	MIoU (%)
U-TAE (weekly)	19.1	9.5	28.7	39.7
3D-Unet (weekly)	17.6	10.2	25.0	37.2
U-TAE (daily)	15.6	7.0	24.2	30.9
3D-Unet (daily)	18.8	11.5	26.1	38.8

The impact of adding extra context can influence the model's performance, either enhancing or diminishing it. Given the strong correlation in daily observations, optimal outcomes are attained through weekly sampling. Moreover, methods employing weekly sampling necessitate fewer computational resources.

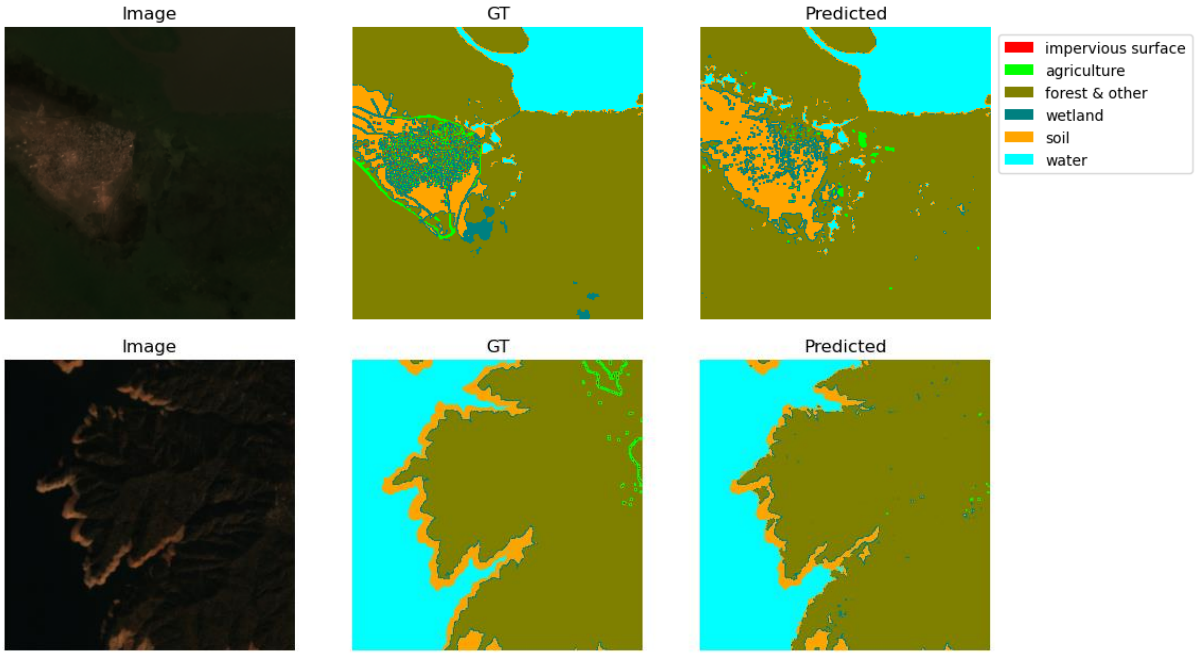

Figure 8: Examples from the DynamicEarthNet dataset [14], each accompanied by its corresponding ground truth semantic segmentation mask and our predicted mask, using the 3D-UNet [20] approach.

In conclusion, our research trajectory will emphasize unsupervised approaches, primarily driven by the fact that our domain involves distinct and undefined anomalies/changes. Given the intricate nature of these anomalies, characterizing them in a supervised manner becomes a challenging endeavour. As a result, our focus on unsupervised methodologies aligns with the inherent complexity of our target changes, allowing us to effectively navigate their detection without predefined labels.

2.4.2 Vegetation monitoring

In the following section we present the implementation details and results of the proposed vegetation monitoring solution.

2.4.2.1 Implementation details

During training, the input data is normalized according to precomputed statistics, in terms of mean and standard deviation, which ensures consistent scale across features, aiding in faster convergence and training stability. We found additional improvement in training stability by normalizing the vegetation height to the range of [-1, 1], and applying hyperbolic tangent nonlinearity to the top of each model.

All models are trained using AdamW optimizer [12] with a learning rate of 0.001 and a weight decay factor of 0.01. Learning rate was decreased by a factor of 10 when the validation loss did not improve for the last 10 epochs. The batch size is set to 32 for all models. We employ data augmentation strategies during training such as horizontal flipping, vertical flipping, and random rotation. No data augmentation is applied during model testing.

We train and evaluate the models using only the valid ground truth measurements. Each data point corresponds to an individual vegetation height measurement tied to a specific location and time of measurement. A data point is deemed invalid under the following conditions:

- The target location lacks a data point,
- The data point lies outside the specified area polygon, which is associated with specific measurement time,
- Data points that overlap with snow or water regions are excluded, as these areas lack vegetation and wouldn't challenge the model.

Another important consideration is treatment of invalid satellite imagery pixels, which occurs due to area polygon cropping of imagery. These pixel values are replaced with the mean of the respective channel.

2.4.2.2 Results and discussion

Quantitative results.

In our assessment, we employed a single-input single-output evaluation. We refrained from utilizing multiple satellite images to generate a series of outputs that would then be averaged for the final result. This approach was chosen to ensure that each prediction can be directly attributed to a specific input, thereby maintaining clarity in our evaluation, and allowing for a straightforward comparison between the predicted and actual outcomes.

We employ the Mean Absolute Error (MAE) metric, which effectively quantifies the average magnitude of errors between predicted and actual vegetation heights. Table 3 shows the results for all tested models on the test split of VHM data. It is evident that UNet architecture outperforms the newer models such as ConvNeXt and Swin. This highlights the fact that cutting-edge models, even though highly sophisticated, are not always guaranteed to provide superior results in all application areas.

Our results are superior to the results reported in [3], which achieved MAE of 2.0 meters on the same target area using the single-input single-output evaluation methodology. Our solution uses more data to train the model, which can contribute towards better performance.

Table 3: MAE results (in meters) for every tested model

Model	DeepLabv3	Swin	ConvNeXt	UNet
MAE [meters]	2.4	2.1	1.9	1.7

Qualitative results.

Figure 9 displays prediction examples from the best-performing model, UNet. The figure provides a tri-fold visualization: the visible spectrum of the original satellite input image on the left, the ground truth vegetation height in the center and the UNet model prediction on the right. We can observe that UNet accurately captures the vegetation height in scenes involving agricultural areas (first row) as well as scenes of urban environment (third row). Note that the satellite imagery also shows only the three visible channels (red, green, and blue). The complete input fed into the model comprises of 12 channels containing raw data, and additional channels calculated on-the-fly using various transformations that aim to provide better data representation.

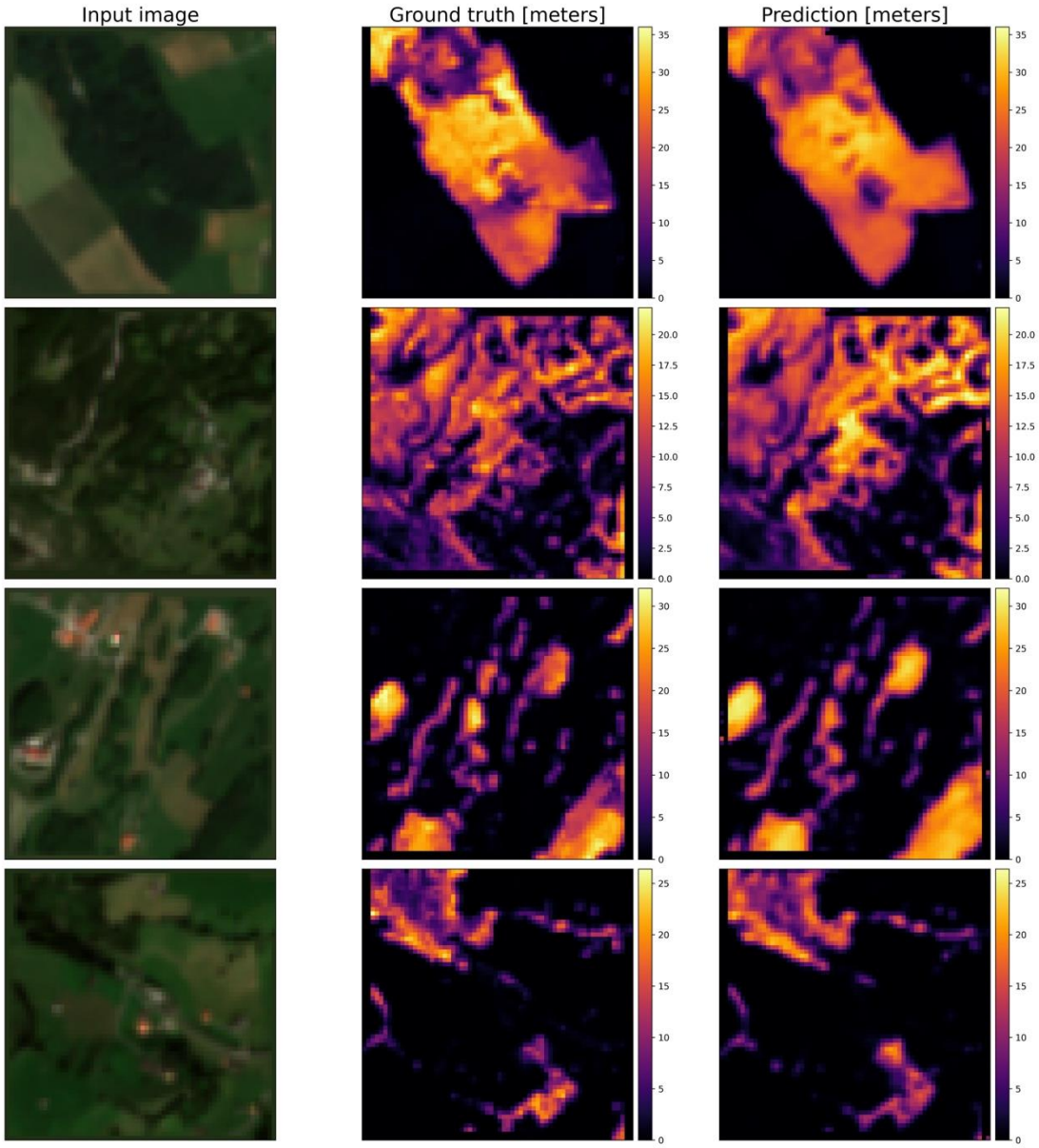

Figure 9: Qualitative results of UNet [4], the best performing model. From left to right the column show the visible spectre of the Sentinel-2 [54] image, the ground truth vegetation height, and the prediction of our model. The resemblance between the ground truth and the model prediction showcases the model's accuracy in capturing vegetation height nuances.

2.4.2.3 Future work

Our future work will focus on the research of self-supervised techniques and advanced data augmentation techniques to further improve the performance of machine-learning models. Furthermore, by receiving information about historic vegetation management events from pilot partners, our evaluation procedure will be more targeted towards specific pilot use cases.

2.5 Deployment

The developed solutions will be made available as-a-service to facilitate easy integrations into user interfaces developed in T7.3 (Section 4) or existing systems available at pilot locations. This scalable and modular design pattern also facilitates re-usability in other AI-based inspection modules (e.g., UAV-based inspection in T7.2 – Section 3). Due to the specific nature of developed solutions, we plan to deploy only inference part of the pipeline. The developed solutions will not have a particular need to be retrained frequently or at all, thus simplifying the deployment pattern and usage of the developed solutions.

The deployment will be facilitated in the following steps:

- **Code packaging:** The inference part of the developed solutions code be packaged using standard Python packaging tools[3], enabling low-level integration with other solutions (e.g., T7.2).
- **Containerisation:** All the developed solutions will be packaged as microservices into containers according to OCI Image Specification[4]. This will enable ease-of-use for other developers, as well as enable deploying the developed solutions in a modular and scalable fashion using container orchestrators (e.g., Kubernetes[5]).
- **AI Model management:** AI model lifecycle will be managed using MLFlow[6]. This will facilitate tracking the performance of the model during an offline training phase, as well versioning of the deployed AI model.
- **REST API:** The solutions will be available as-a-service via REST API with OpenAPI[7] documentation and OAuth 2.0[8] authorization, which will be implemented using FastAPI[9]. This will facilitate high-level integrations with tools developed in T7.3, as well as existing tools available at pilot sites.

The developed solutions will be deployed using cloud-native principles to one of the public cloud providers (e.g., Microsoft Azure Cloud), thus enabling highly reliable and scalable deployment. This approach will be highly beneficial to individual pilots. Deploying to cloud will remove the need for expensive specialized HW and maintenance efforts, related to traditional on-premises deployments. It will also enable precise cost management and effortless scaling, thus making it easier to reach wider adoption of the developed solutions.

[3] https://packaging.python.org/en/latest/
[4] https://opencontainers.org/
[5] https://kubernetes.io/
[6] https://mlflow.org/
[7] https://www.openapis.org/
[8] https://oauth.net/2/
[9] https://fastapi.tiangolo.com/

3 UAV inspection tool

This third section of the document introduces the content related to the tools that comprise the critical infrastructure inspection sub-module using images captured by UAVs. The proposed solutions arise from the efforts within the scope of task T7.2.

The section is structured in a way that gradually introduces content from the general to the specific. It begins with an overview and context of the task at hand, followed by a description of the high-level architecture of the tool. It then delves into a detailed account of the implemented software methods and algorithms, their proof of concept and deployment, and concludes with the hardware composition and integration of the UAV platform.

3.1 General context

The UAV image-based visual inspection module leverages the tremendous technological advances in the field of image analysis to perform targeted, non-invasive examinations of critical infrastructure (CI), its structures and specific elements.

Key innovations in this module include the use of UAV imagery to modernize current visual inspections and the application of AI to automate data analysis. Using state-of-the-art deep learning and computer vision methods, this approach addresses a multitude of problems identified by stakeholders.

Specific visual sensors are used depending on the inspection requirements.

The overall goal is to automate CI preventive maintenance tasks, speed up failure response and create versatile applications for widespread use in different CI scenarios.

These solutions complement satellite inspections, described in the previous chapter 2, with the capacity to analyze much larger areas, providing greater precision and level of detail in specific areas considered high risk or of great importance.

As described in detail in deliverable D7.1, the areas for which this module intends to offer solutions are the inspection of catenary networks and power lines, the detection of generic anomalies (floods, landslides, fire, ...), the inspection of structural or specific components (cracks, corrosion, leaks in pipes, ...), and the 3D virtualization of infrastructures.

3.2 Architecture: high level design

As previously stipulated, this critical infrastructure inspection tool module, which analyzes images captured by UAVs, forms an integral part of the comprehensive inspection module. Thus, its design must be self-contained, modular, and scalable to facilitate seamless integration into the complete solution. A brief definition of these three main design principles, in the context of our specific solution, is included in the following bullet points.

- **Self-Contained Design**: The tool is designed to be self-contained, meaning that it can function independently without relying on other components of the system. This ensures that any changes or updates to other parts of the system do not impact the functionality of the inspection tool.
- **Modular Design**: Modularity is a key aspect of the design, allowing the inspection tool to be easily incorporated into the broader system. This approach enhances the flexibility of the system, enabling the addition, removal, or modification of specific components without disturbing the overall function of the system. This feature is relevant to allow the introduction of new modules during the next phases of the project.
- **Scalable Design**: Scalability is crucial to accommodate potential growth and changes in system requirements. The scalable design of the inspection tool ensures that it can handle an increasing

volume of data or computational requirements, thus future-proofing the system against evolving needs.

To enable this, the developed software application must incorporate an interface that handles communication with the other components of the project, facilitating the exchange of requests and responses. The solution implemented at this initial stage of the project is the deployment of services via a REST API. This RESTful interface ensures standardized communication across different parts of the project, promoting interoperability and enhancing overall system cohesion.

In line with the previously defined design principles of self-containment, modularity, and scalability, the generated code has been dockerized. Dockerization encapsulates the software application within a lightweight, stand-alone, and executable package that includes everything needed to run the application: the code, a runtime environment, libraries, environment variables, and config files. This approach ensures that the application runs uniformly and consistently on any infrastructure. Dockerization, therefore, significantly simplifies the deployment process, allowing for rapid, straightforward access and execution of the software. This encapsulated package can be effortlessly deployed, scaled, and redeployed across diverse environments, further enhancing the flexibility and robustness of the software tool.

Figure 10 shows schematically the elements that compose this architecture, and the way in which they relate to each other. The graphical user interface referred to in the illustration is the one implemented in task T7.3, which is introduced in section 4.

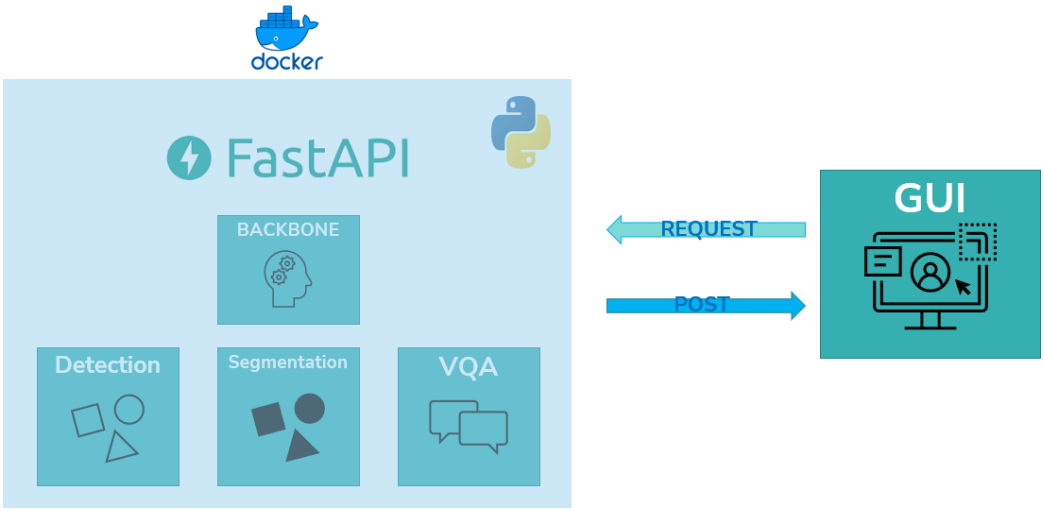

Figure 10: UAV visual inspection tool general architecture.

As reflected in Figure 10, in terms of the code's structure, a main Python script has been implemented to serve as the backbone of the entire process. This script is responsible for coordinating the loading and initiation of the various modules. It also proficiently manages the reception and processing of data in a range of formats, such as directories, videos, individual images, and streaming.

Furthermore, this main script plays a vital role in the integration logic of the modules. Specifically, it ensures that the output from one module can seamlessly become the input for another, creating a harmonious flow of data and operations. Alternatively, if needed, it also has the capability to run the modules autonomously, allowing for flexibility based on specific requirements.

Each of the incorporated modules is designed with a sense of independence. They remain inactive unless they are specifically called upon in the launched application. Adhering to best practices of Object-Oriented Programming, only the necessary module instance is loaded when required. Principles such as the Singleton Pattern [28] have been judiciously employed, guaranteeing that each module is instantiated just once at the beginning to optimize resource utilization.

The implementation of this code structure is designed to try to ensure a system that embodies versatility and efficiency, with the intent of ensuring compatibility across a broad spectrum of hardware devices.

3.3 Tool modules description

Currently, there is a significant upswing in AI models, particularly in the domains of text comprehension and generation, termed **Large Language Models** (LLM), as well as the understanding of visual-textual concepts, known as **Visual Language Models** (VLM). Prominent examples of LLM include Facebook's Llamav2 [29] and OpenAI's GPT-4 [30]. In the VLM category, OpenAI's CLIP [31] and its variations are particularly noteworthy. The rapid technological advancements in these models are undeniable, with leading AI entities such as Microsoft, Facebook, Google, StabilityAI, or OpenAI consistently introducing groundbreaking developments.

In light of this landscape, it is deemed that the most prudent strategy for implement the inspection module is to leverage, adapt, and build upon these pivotal open-source models. Integrating them as foundational elements in more comprehensive frameworks, the aim is to deliver high-quality and valuable solutions tailored for specific infrastructure inspection tasks.

At this juncture of the project, three foundational code modules have been effectively established, setting the groundwork for a progressive expansion of their functionalities in future phases. Their design prioritizes adaptability, ensuring they serve as a robust base for specialized applications planned over the next two years. These modules are:

- **Object Detection and Semantic Segmentation**: This module specializes in detecting and segmenting specific elements within images. It supports both state-of-the-art zero-shot models as well as models meticulously trained for predefined concepts. Furthermore, it offers the ability to perform semantic segmentation of scenes, enhancing their understanding.
- **Visual Question Answering:** An innovative module that interacts with visual content, it furnishes nuanced answers about the image, thereby deepening the comprehension of the embedded visual information.
- **3D Virtualization:** This module capitalizes on image segmentation and leverages the advanced capabilities of the Neural Radiance Field (NERF) algorithm [32]. Through the application of NERF, the module facilitates the transformation of 2D videos into photorealistic 3D scenes, enabling in-depth inspection of these virtualized models.

For the forthcoming phases, there is a proactive plan to incorporate additional modules and augment existing features. On the horizon is the **Synthetic Image Generation tool**, which will leverage advanced models such as Stable Diffusion XL [33]. This addition will facilitate the creation of simulated scenarios, enabling the validation of potential solutions without a sole dependence on real-world imagery during the piloting stages of the project.

Furthermore, discussions are underway about introducing a UAV footage **Change and Anomaly Detection module**. This module would specialize in identifying variances in images captured from the same location but at different temporal intervals. Such capabilities would be invaluable in monitoring environments, infrastructure changes, or any other application requiring temporal image comparisons. As the project evolves, the integration of further modules remains an open prospect, always tailored to meet the evolving requirements of the CI stakeholders.

3.3.1 Object Detection and Semantic Segmentation

The detection and segmentation module is unequivocally the most pivotal and extensive of all modules implemented, both in its current state and as projected for the future. Within the framework of computer vision solutions, tasks related to object detection and semantic image segmentation have historically been the foundational pillars upon which high-value tools are constructed. This holds true for this visual inspection module.

Given the context and aims outlined in the initial two paragraphs of section 3.3, referencing the rise of LLM and VLM models, the semantic detection and segmentation tool integrates various state-of-the-art models. It appends the necessary functionalities and implement distinct solutions jointly.

This module deploys two distinct yet complementary approaches. On the one hand, the developed software aims for a **broad application spectrum**. That is, to seek to detect and segment visual concepts of diverse natures without the need for retraining the models for each specific element, thereby encompassing a wider range of potential use cases. Objects that fit seamlessly into this category include pipelines, electrical isolators, sewers, grates, and other common objects with well-defined shapes and characteristics.

Conversely, in contrast to this need for broad-spectrum models, there exists a demand for the development of **ad-hoc models** for the detection or segmentation of concepts that necessitate specialized handling due to their intricate visual features or owing to their critical nature that demands high performance. Prime examples in this category are fire and smoke. Detecting them poses inherent challenges. Generic models typically offer mediocre performance due to the undefined shape and dynamic evolution of these elements. Moreover, real-time detection with superior performance is paramount to derive actionable insights from the model's output, and in many cases generalist models are larger and require more time to infer a response.

Having delineated this dual approach with clear objectives, and following a thorough review of the current state-of-the-art in these subjects, the integration of up to five distinct models has been achieved to meet the aforementioned needs. The models currently adapted in this module are: Detic, GroundingDINO, X-Decoder, SAM, and YOLOv8. Subsequent paragraphs provide a concise introduction to each of them. It should be noted that while these models build upon previous developments, a decision has been made to abstain from delving into those foundational models in depth, in order to maintain the brevity of the document and ensure readability and comprehensibility.

The first integrated model is **Detic**. Detic [34] was an innovative model upon its release in 2022. It introduced an innovative detection and object segmentation solution that leveraged the capabilities of CLIP (Contrastive Language–Image Pre-training) [31] to merge text and image understanding. This method allows Detic to adapt to detection tasks using datasets primarily designed for classification, tapping into their extensive labeled resources. Remarkably, with the integration of CLIP, Detic can recognize objects without any prior exposure during its training, showcasing its zero-shot detection capabilities, and can detect any concept/object given input as plain text (open-vocabulary). Facebook is the main entity associated with the development of this model.

Similarly, **GroundingDINO** [35] is also an open-set object detector. This means it can be instructed on which concepts to locate in an image through a textual input, and it does not require the item to be within a predefined class list for the detector to function correctly. This model merges a detector based on the Transformer architecture, DINO, with grounded pre-training. While it shares some similarities with Detic's approach, it employs distinct methodologies. Moreover, it is worth noting that GroundingDINO is from a more recent publication, dated March 2023, and it is associated to Microsoft researchers.

Both models demonstrate impressive performance on the COCO open-vocabulary benchmark [36], a common metric highlighted in both papers. GroundingDINO slightly outperforms Detic in zero-shot AP, scoring 46.7 mAP compared to Detic's 45.0 mAP. However, it is essential to understand that these metrics are only roughly comparable. Differences in the methodologies and approaches of the two models can influence the outcomes, underscoring the value of implementing and testing both solutions.

Also developed by researchers at Microsoft in collaboration with other institutions, **X-Decoder** [37] is a sophisticated model designed to identify which object/cluster in an image each pixel belongs to and determine its corresponding language token. Built on an encoder-decoder framework, this model employs an image encoder to capture visual features and a text encoder to process language-based

queries. The decoder then uses this information to predict specific image segments and their related language semantics. X-Decoder seamlessly integrates a range of image segmentation and vision-language tasks, achieving top-tier results in open-vocabulary segmentation. This includes tasks like scene semantic segmentation (e.g., distinguishing houses, trees, clouds, sky, and ground in an image) and referring segmentation tasks (e.g., identifying the closest house in an image). This work was published in December 2022. In illustrating the capabilities of this model, Figure 11 shows the performance of the open-vocabulary X-Decoder in various scenarios relevant to the project context.

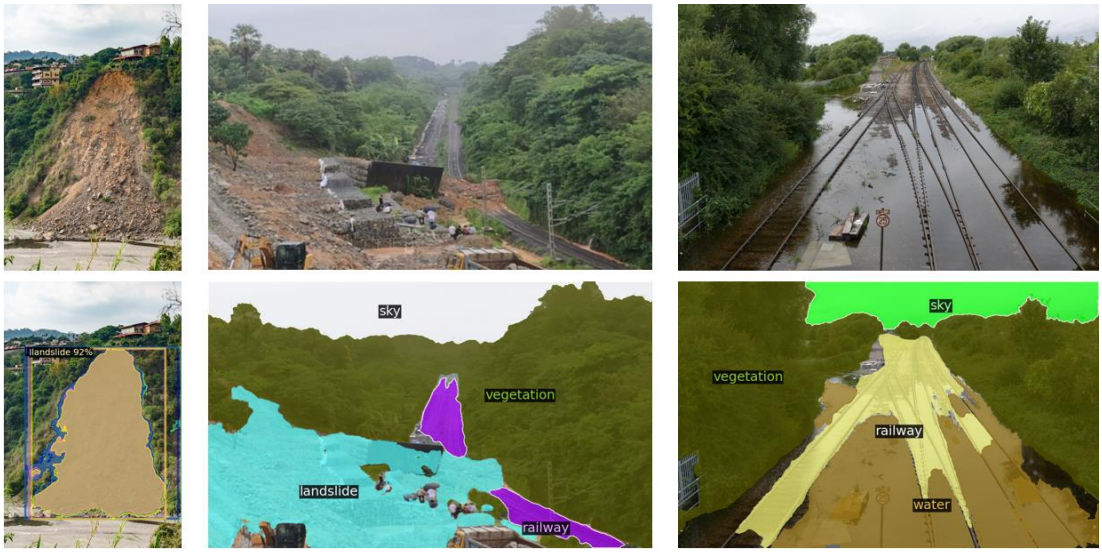

Figure 11: Landslide and flood detection/semantic-segmentation using X-Decoder on aerial imagery. Source: [58] (left); [59] (centre); [60] (right).

The third model introduced is **Segment Anything Model** (SAM) [38]. As highlighted in the publication paper of SAM, X-Decoder played a partial role in inspiring this new design, which was released in April 2023 by researchers from Microsoft. Similar to its predecessor, SAM is a highly versatile image segmentation model that leverages cues such as text or specific image positions to effectively segment concepts across a broad range of segmentation tasks. In the months following the original SAM model's release, the community has introduced enhancements to the base model, leading to the development of SAM-HQ [39], which outperforms SAM across the nine datasets evaluated in the study. For reference, and while acknowledging that methodologies in each study can influence outcomes, one can compare the metrics of the three models on the COCO dataset [36]. X-Decoder reports the lowest mAP at 40.5, followed by SAM at 48.5, and SAM-HQ leading with 49.5. Despite this comparison, it's important to note that, based on our experimental implementations and tests, SAM-HQ distinctly excels in segmenting specific objects and elements. However, X-Decoder demonstrates superior scene comprehension, which could be invaluable for future inspection applications.

Table 4 provides a summary comparison of the models introduced thus far, based on the results presented by their respective authors. Section 3.4.1 will introduce application examples of the implemented models to offer a qualitative assessment of the outcomes.

Table 4: Comparison of mAP metrics for models in COCO dataset across Detection and Semantic Segmentation tasks. For detection, mAP evaluates bounding box accuracy; for segmentation, it gauges the precision of segmentation masks.

Models	Detection	Semantic Segmentation
DETIC	45.0	-
GroundingDINO	46.7	-
X-Decoder	-	40.5
SAM	-	48.5
SAM-HQ	-	49.6

It is essential to emphasize that the four models introduced thus far are of universal applicability, endowed with open-vocabulary zero-shot detection and segmentation capabilities, without the need for retraining or select the objects to be detected from a closed list of classes. Furthermore, they operate under an open-vocabulary approach, without being restricted to predefined lists of identifiable objects. These attributes make them highly versatile tools, suitable for integration into more complex workflows as the project progresses.

Finally, having introduced the models responsible for delivering the previously mentioned broad-spectrum solutions, it remains to specify the foundation for specialized, ad-hoc solutions. For such applications, **YOLOv8** stands out as the prevailing state-of-the-art model, making it the chosen implementation. While there is not an official paper detailing YOLOv8 as of this document's date, comprehensive insights into its features and architecture can be found on [40], the GitHub repository of its developer, Ultralytics. Drawing from the legacy of the "You Only Look Once" family [41], YOLOv8 incorporates enhancements and novel features to this series of real-time object detection systems, that employs a singular convolutional neural network (CNN) to process the entire image, predicting object locations and classifications. This method differs from other detectors, instead of proposing regions and subsequently classifying them (two-steps detectors), YOLO segments the image into grids, predicting bounding boxes and class probabilities for each grid cell in one go. This streamlined approach grants YOLO its remarkable speed and efficiency, setting it apart from other detection methods and making it perfect for this ad-hoc model detection and segmentation solutions.

More examples of the performance of these models in specific use cases of the project are given in section 3.4.1, where some PoCs are introduced.

3.3.2 VQA

The Visual Question Answering (VQA) module, like the previous module, aims to extract the maximum amount of information from images by leveraging the latest advancements in LLM and VLM. VQA stands at the cutting-edge intersection of computer vision and natural language processing. Within a VQA system, an image is processed in tandem with a text-based query pertaining to its content. This system harnesses the power of convolutional neural networks for image analysis and either recurrent neural networks or Transformers for textual understanding, producing a contextually appropriate response to the given question. This approach offers not only precise identification of visual components in the image but also a profound semantic comprehension to link the visual content with the linguistic inquiry.

To implement this module, the chosen tool is **BLIP-2**. BLIP-2 [42] is an innovative pre-training strategy that bridges the gap between vision and language by integrating pre-trained image encoders with large language models. Integral to BLIP-2 is the Q-Former, a component specifically designed for modulation. While BLIP-2 integrates two robust pre-trained models (visual and linguistic), the Q-Former refines their interaction, ensuring optimal synchronization. A salient feature of BLIP-2 is its adaptability; as advancements in model architectures emerge, BLIP-2 can be updated to incorporate these

enhancements, maintaining its position at the technological vanguard, by only fine-tuning the Q-Former instead of several huge networks. Empirical evaluations have underscored BLIP-2's superiority over other leading models, such as Flamingo[43]. In benchmark tests, specifically in zero-shot VQAv2 tasks [44], BLIP-2 surpassed the Flamingo80B model by a margin of 8.7%.

For a more tangible understanding, Figure 12 provides a visual representation of the model's capability to interpret scenes and address specific queries about them. This visualization underscores the efficacy and potential of the chosen approach. Further exemplifications and use-case evaluations are detailed in section 3.4.2.

{"what do you see?": ["a rope attached to the side of a waterfall"], "are there cracks in the concrete?": ["yes"], "where are cracks in the wall?": ["in the middle of the wall"], "the metallic rope is broken?": ["no, it's not broken, it's just rusty"], "the metallic rope is in good condition?": ["yes"], "Is the grill clogged?": ["no"], "Are there any people?": ["no"]}

Figure 12: VQA example of BLIP2 image interrogation. Above: original image taken in HDE's installations. Down: examples of queries and answers provided by the model.

3.3.3 3D Virtualization

The third module introduced is a 3D reconstruction and virtualization module, to obtain 3D point clouds of the infrastructure with inspection purposes. The foundation of this module lies in the innovative technology of **Neural Radiance Fields** (NeRF) [32]. This technique is at the leading-edge computer vision advancements, harnessing the power of deep neural networks to transform two-dimensional images into intricate three-dimensional scenes. NVIDIA, a vanguard in the tech sphere, has further refined this approach with their iteration termed Instant-NeRF [45]. This enhancement not only expedites the NeRF procedure but also optimizes the creation of neural radiance fields, ensuring a more cohesive and agile process.

An added key component of NVIDIA's approach is the use of the **COLMAP** tool [46], a robust utility that enables dense scene reconstruction from a collection of images. While this tool greatly bolsters the reconstruction phase, there are instances where the output may exhibit noise or other imperfections. To counteract these anomalies, the sophisticated segmentation models previously detailed can be invoked. Utilizing these models allows for the precise elimination of any scene noise, culminating in the production of immaculate and precise three-dimensional depictions of the targeted objects.

In summary, the synergy of NeRF technology, NVIDIA's Instant-NeRF enhancements, the capabilities of COLMAP, and the accuracy of state-of-the-art segmentation models collectively set a gold standard

for the fidelity and precision of 3D scene reconstructions, and thus are the processes implemented in the proposed virtualization pipeline.

The steps followed and an example of the results obtained by proceeding in this way can be visualized in Figure 13. A PoC with real data taken with UAV in the context of this Pilot 0 is attached in section 3.4.3.

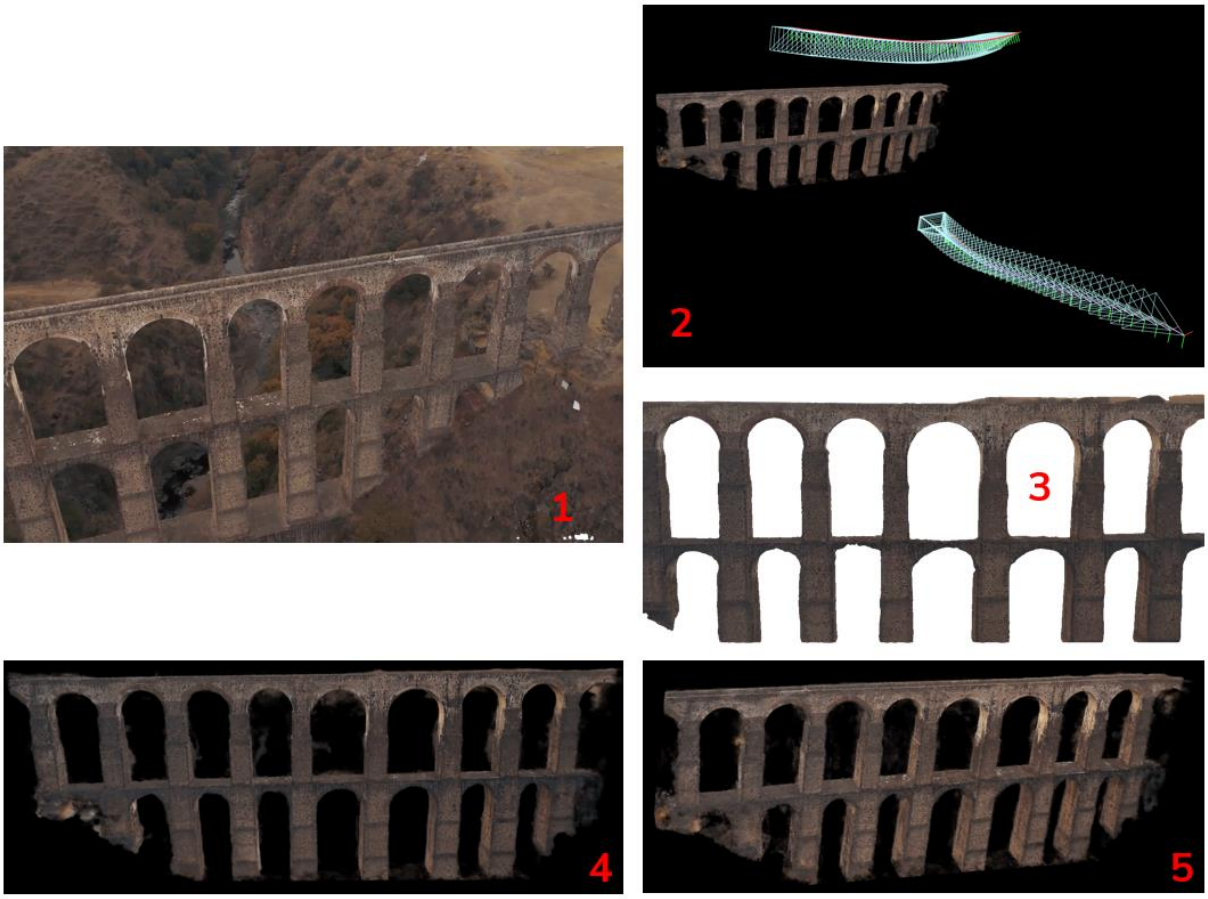

Figure 13: 3D virtualization of an aqueduct from a UAV footage video. Step 1: collect raw images of an infrastructure; Step 2: use COLMAP to generate camera path; Step 3: extract background with SAM-HQ; Step 4-5: NeRF training and 3D model visualization. Source: [61].

3.4 Tool modules lab validation

As highlighted in preceding sections, validation of the proposed UAV platform and pipelines within real-world scenarios of critical infrastructure is earmarked for Pilots 1 and 2, set for execution in 2024 and 2025. In the context of Pilot 0, validation is confined to a comparative analysis of metrics from the conducted state-of-the-art study, own metrics in the case of YOLOV8, and a few PoCs that have been undertaken using data provided directly by CI entities or from open online sources.

The ultimate objective of this validation is to demonstrate that the implemented models (section 3.3) can serve as a valuable component or even as the entirety of an image analysis process for inspecting the elements of interest.

3.4.1 Object Detection and Semantic Segmentation

Regarding the detection and segmentation module, a substantial number of tests have been conducted to qualitatively verify that the implemented models perform well in scenarios and situations encompassed within the project.

Firstly, Figure 14 includes several examples of object detection using the GroundingDINO open-vocabulary model. This model provides the bounding box of the object specified as textual input, with the words "grate, manhole, insulator" being used in this instance.

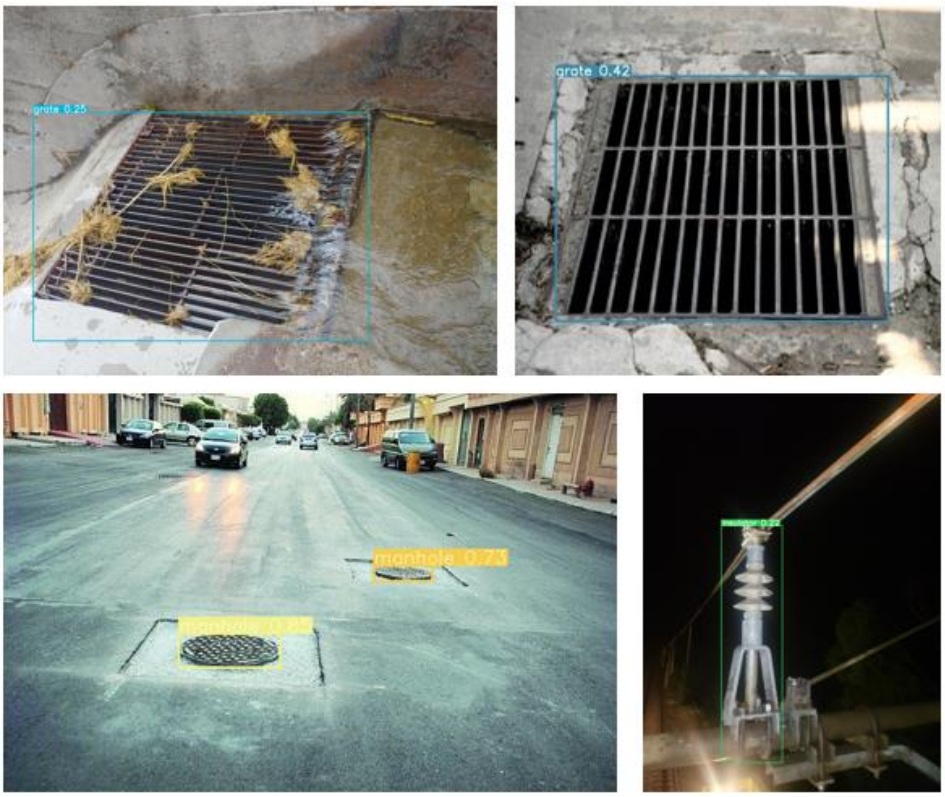

Figure 14. GroundingDINO detections with open-vocabulary: grate, manhole, isolator. Source: [62] (down-left); CI stakeholders (others).

Secondly, Figure 15 displays the segmentation of images of pipes that could be part of ACO's installations. This test showcases the outcome of using the word "pipe" as input to the X-Decoder open-vocabulary model.

Figure 15: Pipes semantic segmentation results with X-Decoder open-vocabulary: pipe. Source: [63] (left); [64] (centre); [64](right).

The images derived from this segmentation model can serve as input for a subsequent model focused on detecting leaks or corrosion in the pipes. The extraction of background noise simplifies the challenge and aids the tasks of subsequent models applied to the output image.

A second test, based on the same concept of employing semantic segmentation to eliminate image noise, can be observed in Figure 16. On this occasion, the segmentation results from utilizing GroundingDINO to identify each object, followed by SAM-HQ to segment each pixel. In addition to the original images, the textual input to the model is "pylon, insulator".

Figure 16: Background extraction with GroundedSAM (GroundingDINO + SAM) open-vocabulary: pylon, insulator. Above: original images; Down: SAM segmentation results. Source: CI stakeholders (left); [66] (left-centre); [67] (right-centre); [68] (right).

Lastly, regarding the ad-hoc models trained for specific tasks, YOLOv8 has been trained for fire and smoke detection using a dataset of images generated through Stable-Diffusion v2.1. This dataset originates from another European project, Sylvanus. Figure 17 displays mAP metrics achieved during the training of this model, reaching 0.64 in mAP.

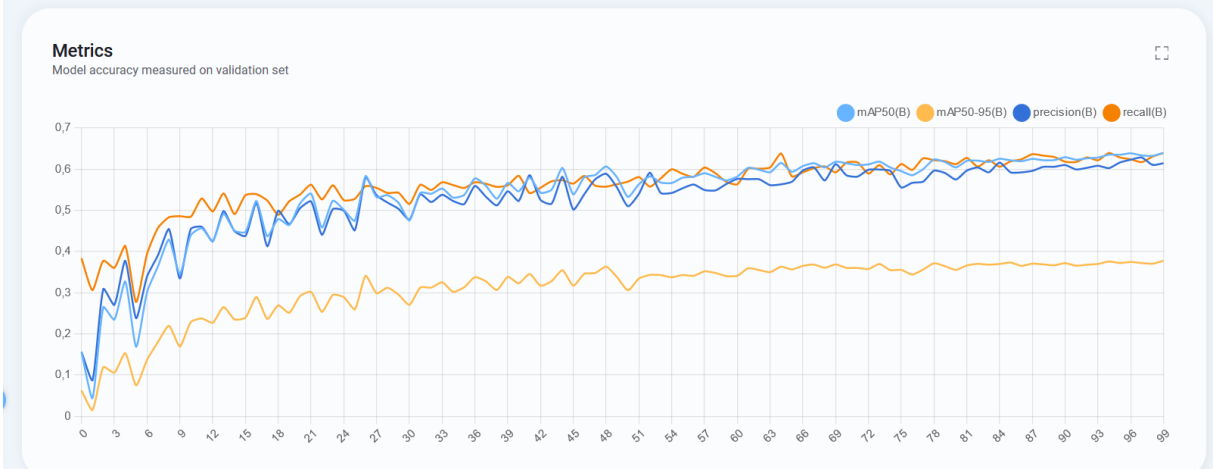

Figure 17: YOLOv8 fire and smoke detector mAP metrics.

Figure 18 showcases some results obtained by inferring with the model on images produced using generative image models, with the intent of testing the model in a context akin to real-world application. As can be observed, the actual challenge to address in this instance is the detection from UAVs of potential fires caused by sparks emitted from trains during braking.

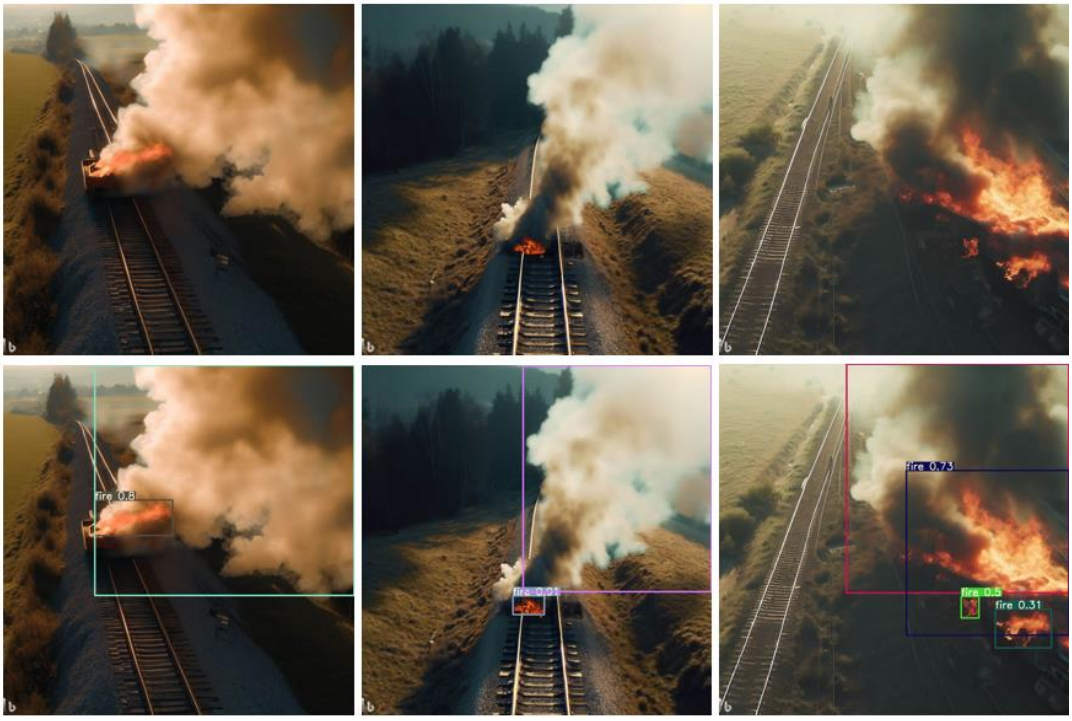

Figure 18: Fire and smoke detection with Yolov8 model over Microsoft Bing AI generated images.

3.4.2 VQA

The PoC conducted with the BLIP-2 model aims to ascertain whether such models can offer pertinent information for decision-making regarding the condition of specific elements within facilities or structures. This approach could potentially obviate the need for training distinct classifiers for each element to be inspected, thereby encompassing a broader range of use cases.

The subsequent illustrations depict clear examples of how this model can assist in identifying damages or issues in elements and scenarios pertinent to this project. Figure 19 demonstrates its application in assessing the integrity of three pipes with varying degrees of deterioration, providing valuable insights both in the textual output and in the similarity percentage values between the queries and the input images.

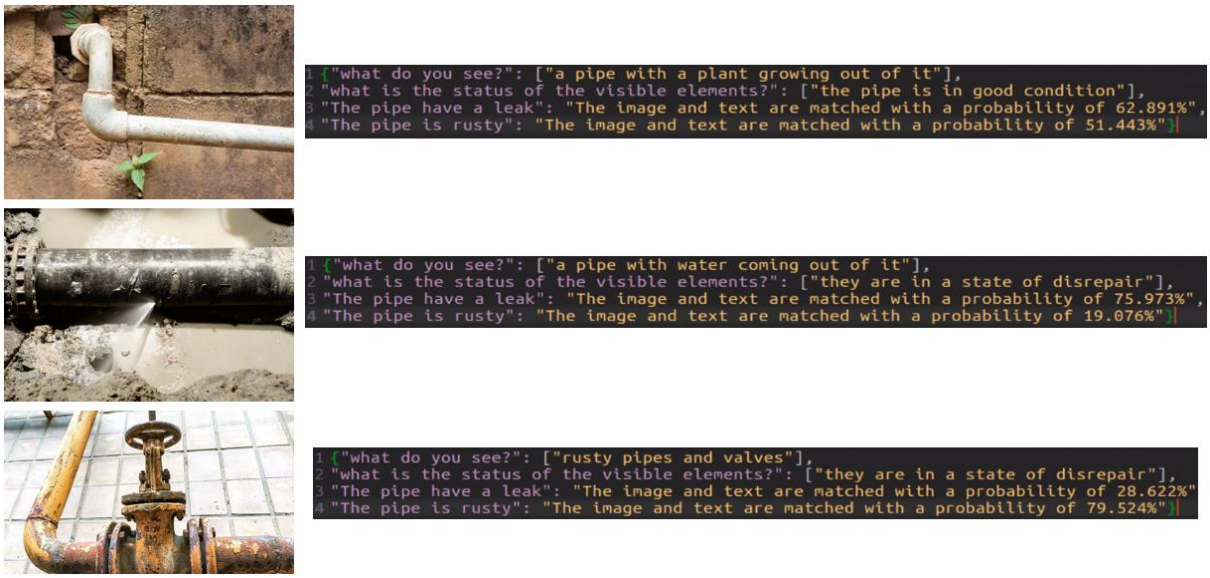

Figure 19: BLIP-2 pipe status image interrogation. Source: [63] (up); [64] (centre); [65] (down).

Figure 20 presents positive outcomes in the endeavor to replace image classifiers with these types of textual verifications, accurately determining the maintenance status of the ceramic insulators in power lines.

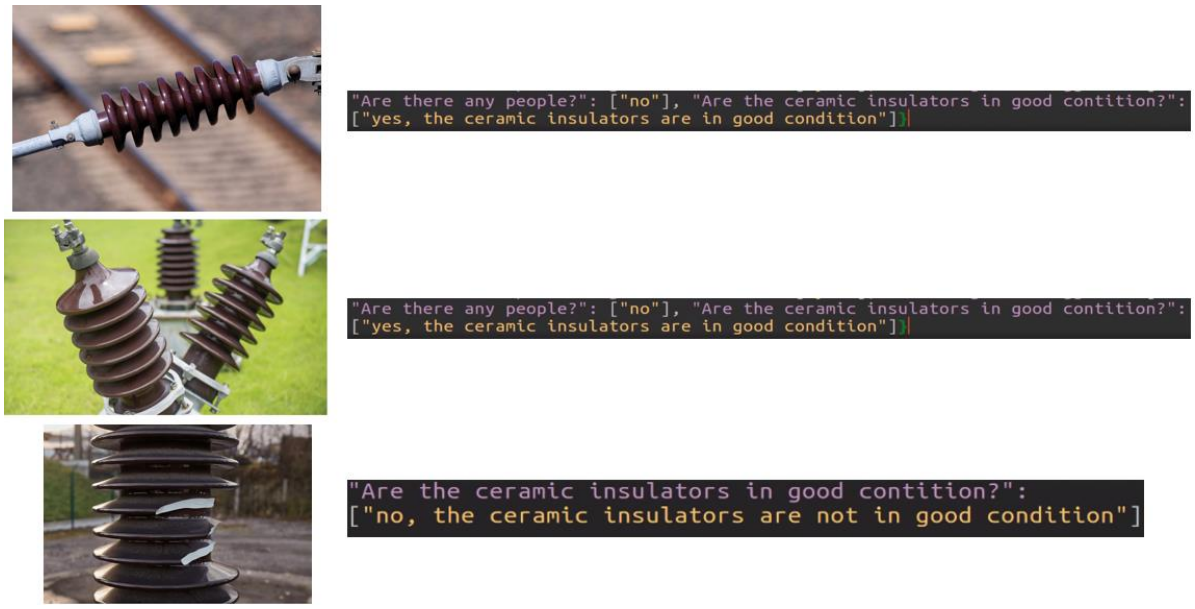

Figure 20: BLIP-2 insulators status image interrogation. Source: [66] (up); [67] (centre); [68] (down).

In the last of the three illustrations, Figure 21, it can be observed how a straightforward pipeline combining detection, segmentation, and the VQA model can address challenges such as automatically verifying the blockage or occlusion of grates in remote locations.

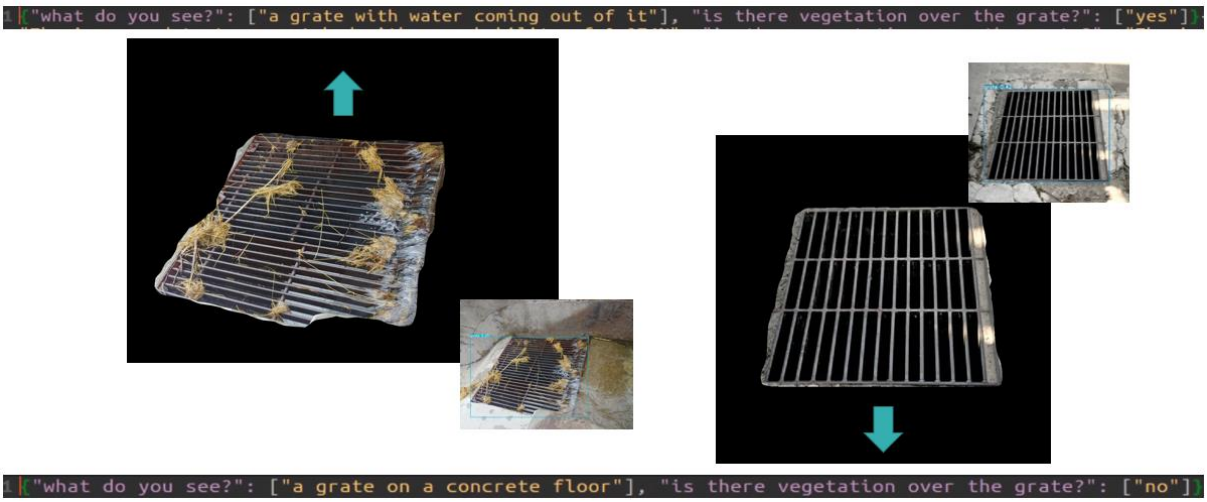

Figure 21: BLIP-2 grate clogging status image interrogation.

3.4.3 3D Virtualization

The PoC conducted to validate the 3D model generation pipeline, integrating NeRF, COLMAP, and SAM-HQ, incorporated the recording of a Point Of Interest (POI) video using a recreational UAV. POI flights involve executing a circular route around an object or structure intended for 3D reconstruction. Figure 22 displays an example of the kind of images included in the recorded video, in which two laps at varying altitudes are taken around a grain silo.

Figure 22: Original UAV footage of a silo, video frame example.

From the original images, COLMAP is employed to extract the path and pose of the camera for each shot, as depicted in Figure 23. Also, in this picture it is possible to notice the segmentation process carried by SAM-HQ.

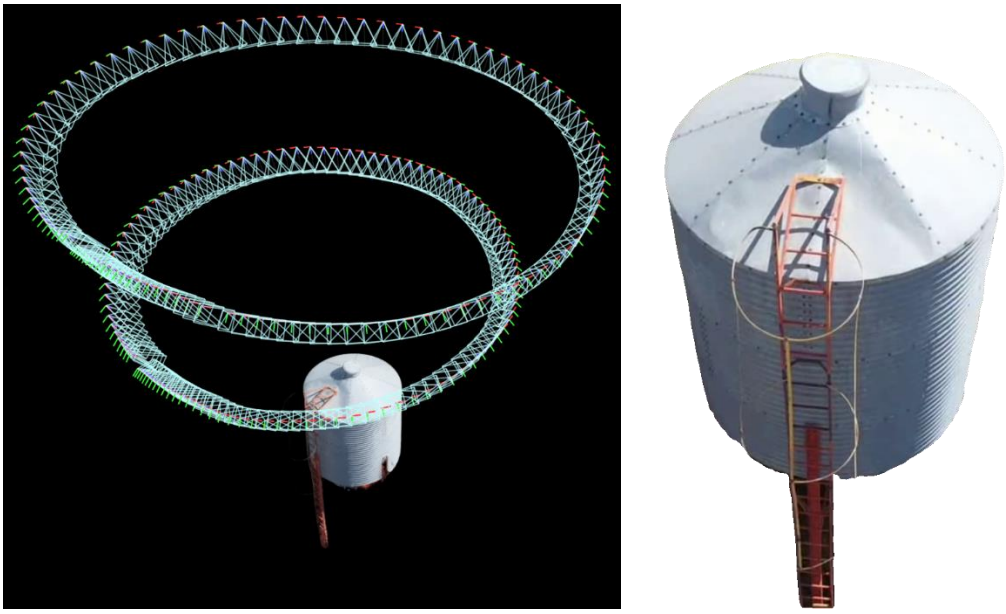

Figure 23: Left: UAV path reconstruction with COLMAP, instant-ngp GUI screenshot. Rigth: SAM-HQ silo segmentation background extraction.

Ultimately, with the segmented images and the camera path as inputs, the NeRF model is trained. The subsequent captures presented in Figure 24 display various perspectives of the reconstructed model, where one can clearly discern the dents in the silo and its overall condition.

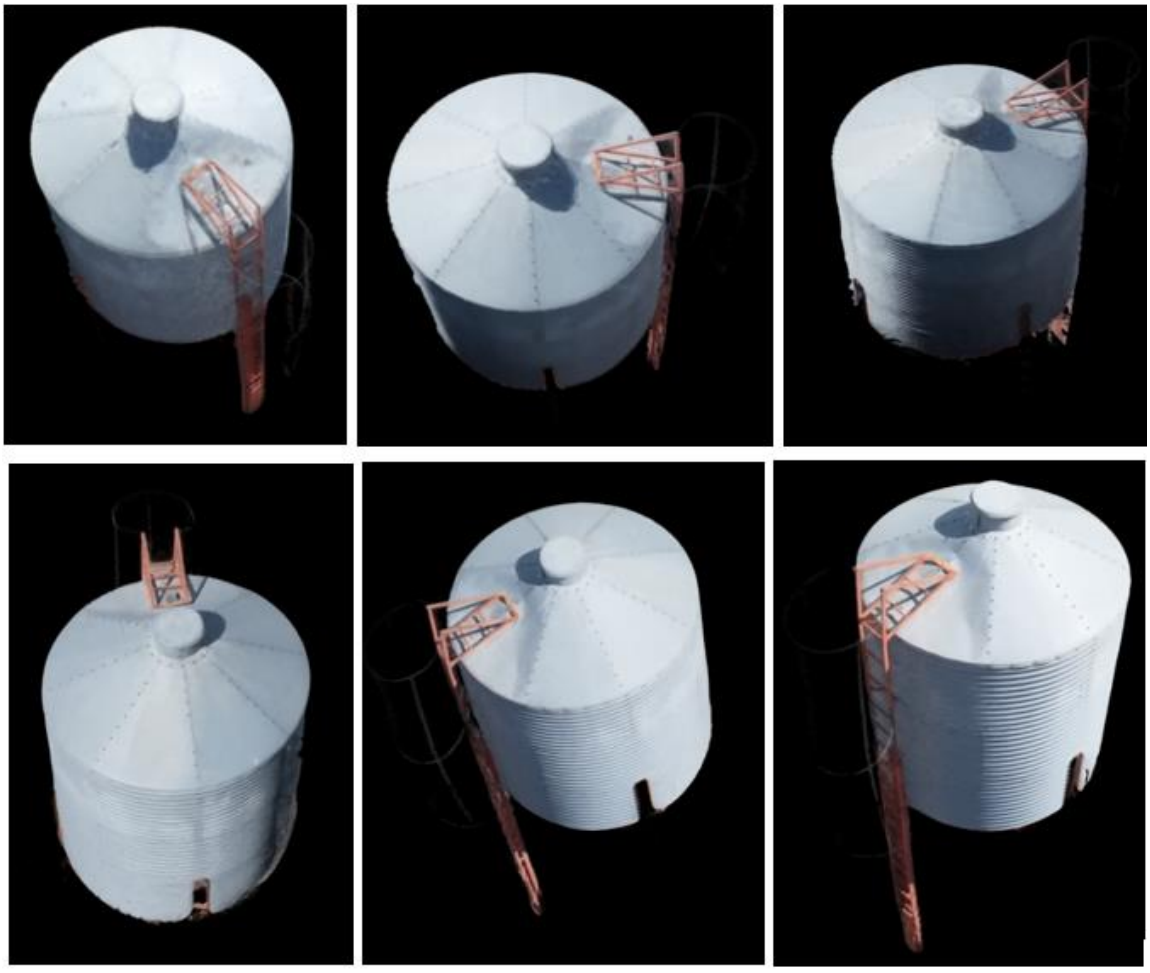

Figure 24: GroundedSAM and instant-ngp 3D silo virtualization, instant-ngp GUI screenshots.

3.5 Deployment

As introduced in 3.2, the software tool under development will be deployed in two distinct modes, depending on the specific requirements of each use case. The deployment strategies include a REST API service and as embedded software on the UAV onboard card.

- **REST API Service Deployment**: This mode of deployment is designed for situations where the software tool does not need to be run in real-time. The software tool will be structured as a REST API service, providing a set of clearly defined methods of communication. It will utilize standard HTTP protocols, making it universally accessible across the network. This type of deployment allows for interaction with other software components, enabling the exchange of information (images and videos) and commands between the tool and other systems or actors, like CI stakeholders.
- **Embedded Software Deployment**: In use cases where direct integration with the UAV is required, the software tool will be deployed as embedded software on the UAV onboard card. This deployment strategy is best suited for instances requiring low latency, high performance, and direct control over the UAVs functionalities. The software will be customized to operate within the specific hardware constraints of the UAV onboard card, ensuring optimal performance and reliability. This method allows for real-time processing and response, which is critical in scenarios involving immediate decision-making based on the UAVs sensor data.

Both deployment strategies aim to provide a flexible, robust, and secure solution that can be tailored to meet the diverse needs of different use cases. The choice between the two will be dictated by the specific operational requirements, technical constraints, and security considerations of each case. This dual approach underscores our commitment to developing a versatile tool that can be seamlessly integrated into a wide range of operational contexts.

3.6 UAV platform lab integration

The UAV platform is a flying tool that approaches by air the facilities of interest, thus bypassing the restrictions imposed by ground-based transition on them. As shown in the following Figure 25, the UAV platform consists of three main parts:

- The aerial vehicle that transits to the inspection point/site,
- The camera that is the eyes for structure's inspection,
- The microcomputer, which is the brain of the tool. This unit runs an application that processes the images received/captured by the camera and is able to detect a number of problems in the inspected facilities.

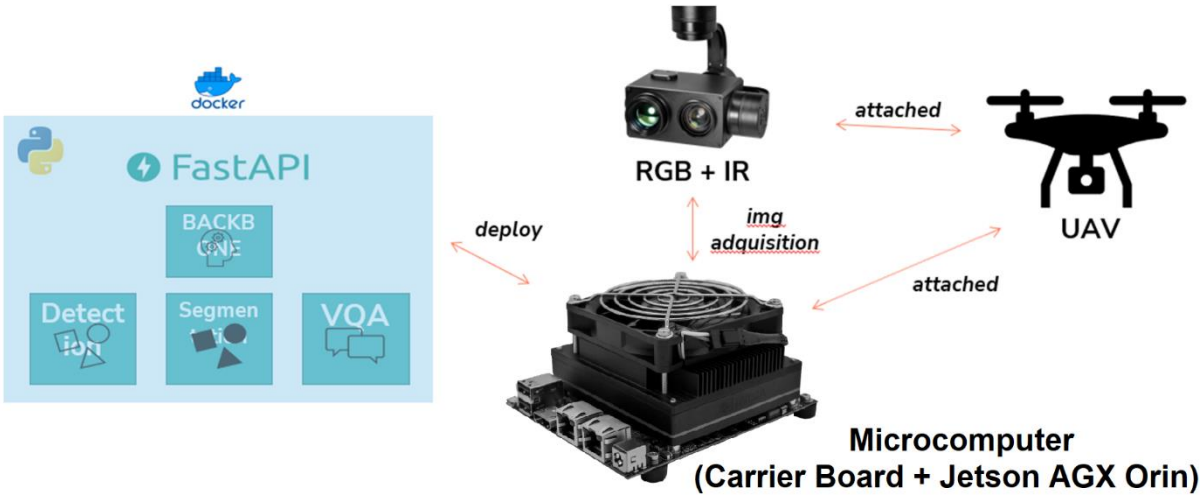

Figure 25: UAV platform components.

3.6.1 Hardware Specifications

The hardware that UAV Platform tool consists of is of high specification for the most effective performance for its intended purpose. The following is a description of those specifications for each part of the tool according to the above segmentation.

- **The Aerial Vehicle,** of the tool is a UAV and more specifically the ATLAS 204 N22 model [47], shown in Figure 26. This system is designed to deliver high reliability & mission oriented multi-rotor capabilities in the fields of defense, security & industrial surveillance applications. It follows the highest industry standards, with state-of-the-art mission command systems, encrypted RF links, redundant security systems and a configurable payload node. One of its most important features is that it is able to operate under harsh environmental & electromagnetic conditions and with minimal human power using advanced mission-oriented algorithms.

Figure 26: UAV Atlas 204 N22 Model [47].

- When compact and folded, the dimensions of this device read 23 x 23 x 40 cm, embodying portability, and ease of transport. Upon expansion, it extends to 63 x 63 x 40 cm, unveiling its larger operational configuration.
- This device bears a rated weight of 8.7 kg, accommodating the dual camera load with exceptional balance. Its prowess extends to a maximum take-off weight of 11.7 kg, demonstrating its capacity for carrying substantial payloads.
- Flight autonomy offering an impressive span of 55 to 70 minutes in the air. This prolonged flight duration ensures ample time for complex tasks and missions.

- Telemetry and video transmission range excel, spanning a distance of 15 km to 20 km in Line of Sight (LOS) conditions. This extensive range opens avenues for expansive exploration and surveillance.
- Ensuring resilience, the device boasts an IP43 rating for tightness, safeguarding its internal components against dust and water ingress.
- Ascending to heights of 9,000 ft AMSL (3,000 m) is well within the device's capabilities. Additionally, it confronts wind speeds of up to 15 m/s (Up to 7 Beaufort scale[10]), showcasing its stability and control even in challenging conditions.
- Velocity reaches new heights as well, achieving a maximum flight speed of 23 m/s (82 km/h), making swift aerial maneuvers attainable.
- The device relies on GNSS (GPS & GLONASS) for positioning, ensuring accurate location data for precise navigation.
- Secure communications are maintained through a frequency of 2.4 GHz, fortified with AES 128 & 256 Encryption, safeguarding data transmission.
- Operating temperature range spans from -10 °C to +50 °C, allowing the device to function effectively across various climates.
- The device empowers users with automatic take-off and landing capabilities, managed by its sophisticated autopilot system.
- Its power source consists of two Lithium batteries, each rated at 22.2V and 22,000mAh, providing the energy required for sustained flights.
- For enhanced visibility, navigation lights in green, red, and white illuminate its path, ensuring safety and situational awareness.
- The device's capabilities extend to payloads, accommodating Dual EO/IR/LRF payloads for versatile data collection and analysis.

In conclusion, this device epitomizes technological excellence, blending compactness with expansive capabilities. Its dimensions, endurance, communication prowess, navigational finesse, and payload versatility converge to create a device ready to redefine exploration, surveillance, and data gathering across a diverse spectrum of applications.

- **Camera 1**: The primary camera on the instrument is an EO/IR (Electro-Optical/Infra-Red) imaging mechanism, encompassing both standard visual and infrared detection capabilities. As they cover both the visible and infrared spectrum, EO/IR mechanisms ensure complete situational recognition during day, night, and in dimly lit scenarios. Distinct attributes of EO/IR setups include their capability for distant image capture and image steadiness. They should possess the proficiency to discern and track dynamic targets, even amidst adverse environmental scenarios. For the current tool, we've opted for the **Z10TIR** model, shown in Figure 27. All information regarding their characteristics can be found in [48], and is the source of the data provided below.

[10] The Beaufort scale is an empirical measure that relates wind speed to observed conditions at sea or on land. Its full name is the Beaufort wind force scale.

Figure 27: Camera Z10TIR [48].

Generally, the Z10TIR system is examined through three technical specifications parameters: Stabilization, EO camera and IR thermal imager.

Stabilizing: The Z10TIR is mounted on an advanced 3-axis gimbal, offering precision motor rotation with a control accuracy of ±0.02°, driven by a specialized processor. Instead of the typical electrical slip rings found in many gimbals, this model employs a distinct mechanically restricted design with hidden wiring, enhancing data transmission stability and longevity. Vibration is effectively countered with four damping balls and a lightweight damping plate, ensuring smooth video capture. The gimbal allows for a full 360° rotation. This design ensures that clear and steady footage is achieved, even during high-speed UAV flights.

EO camera: The EO camera features a 1/3" CMOS sensor, boasting a color sensitivity of 0.5lux@F1.6, 2.48 million effective pixels, and delivers 1080p HD image clarity. This, along with superior optical zoom and rapid autofocus, is tailored for UAV-based photography. The Z10TIR integrates a precision ULIX thermal sensor from France, optimized for uncooled long-wave (8µm ~ 14µm) imaging with a 25mm lens. This enables simultaneous capture and transmission of thermal and visible images. Users can choose between two thermal resolutions: the standard 640x480 and a lower-tier 384x288 option. The ULIX sensor uncovers hidden thermal details, making temperature variations discernible. Such insights can highlight structural damages, and offer other crucial information often unseen by the human eye.

IR thermal imager: The built-in IR thermal imager incorporates advanced algorithms for normalization, cross-correlation, and tracking. Paired with a recapture mechanism for lost objects, it ensures consistent target tracking. It also offers customizable on-screen display (OSD) features like adaptive gating, crosshairs, and trace information. The system can track at speeds up to 32 pixels per frame and covers object sizes ranging from 16x16 pixels to 160x160 pixels. With a minimum signal-to-noise ratio (SNR) of 4dB and position pulse noise values averaging below 0.5 pixel, the imager's precision and tracking performance are significantly enhanced.

Some additional H/W parameters for this unit is following:

- Working voltage **12V.**
- Dynamic current **800~1000mA @ 12V.**
- Idle current **800mA @ 12V.**
- Working environ.tmp. **-20℃ ~ +60℃.**
- Output **microHDMI(FHD output 1080P 30fps) / IP(1080P/720p, 25/30fps)/Skyport.**
- Local-storage **TF card (Up to 128G, class 10, FAT32 or ex FAT format).**
- Photo storage format **JPG (1920*1080/1280*720).**
- Video storage format **MP4 (1080P/720P 25fps/30fps).**
- Control method **PWM / TTL / S.BUS/ TCP (IP output version /Skyport).**

- **Camera 2**, of the tool is a LiDAR (Light Detection and Ranging) camera, also known as a LiDAR sensor or LiDAR scanner, is a remote sensing technology that uses laser light to measure distances and create detailed 3D maps or point clouds of objects, environments, and surfaces. It works on the principle of emitting laser pulses and measuring the time it takes for the pulses to reflect back from objects. LiDAR cameras are commonly used for various applications, including autonomous vehicles, mapping, surveying, forestry, archaeology, and more.

  LiDAR cameras emit short laser pulses of light in various wavelengths, often in the near-infrared range. These pulses are directed toward the target surface or object. By measuring the time it takes for the laser pulse to reflect back to the LiDAR sensor, the distance between the sensor and the target can be calculated using the speed of light. LiDAR sensors can capture multiple returns from a single laser pulse, allowing them to create detailed profiles of objects and surfaces, including multiple layers within vegetation, buildings, and terrain. The collected distance measurements are used to create a point cloud, which is a three-dimensional representation of the objects and surfaces in the sensor's field of view. Each point in the cloud represents a specific location in space and is accompanied by its x, y, and z coordinates.

  LiDAR cameras can generate highly accurate 3D maps of landscapes, buildings, and other environments. In the context of SUNRISE, it is a very critical tool in calculating the vegetation status where this must be controlled.

  For the current tool the **GS-100C+** has been selected. Figure 28 displays the specific camera followed by its specifications. Specifications come from [49] .

Figure 28: GS-100C+ Camera [49].

System (GS-100C+ Lidar camera) Specification:

System Parameters

- Accuracy	**≤10cm@110m.**	- Dimension	**15.5*9.2*9.3cm.**
- Weight	**1036g Storage 64 GB.**	- Max support	**128GB TF card.**
- Working Temperature	**-20°~＋55°.**	- Carrying Platform	**Multi Rotor/VTOL.**

Laser Unit

- Measuring Range **190m@10%.**
- FOV **70°the cirular view.**
- Laser Class **905nm Class1 (IEC 60825-1:2014).**
- Range Accuracy **(1σ @ 20m) 2 cm.**
- Laser Line Number **Equivalent to 64-beam.**
- Data **Triple echo,720,000 Points/Sec.**

POS Unit

- Update Frequency **200HZ Position Accuracy ≤0.05m.**
- GNSS Signal Type **GPS L1/L2/L5, GLONASS L1/L2.** and **BDS B1/B2/B3, GAL E1/E5a/E5b.**
- Pitch /Roll Accuracy **0.015°.**
- Heading Accuracy **0.040°.**

Camera

- FOV **80°.**
- Effective Pixel **24 MP.**
- Focal Length (mm) **15.**

Operation Efficiency Table

Flight Height (m)	**Accuracy**	**Single Flight Operation(km²)**
50	≤5cm	0.88
70	≤7cm	1.28
110	≤10cm	1.92

- **The Microcomputer**, of the tool is a small dimensional computer with a total weight of 1.6 Kg and X-Y dimensions of 125 x 104.6 mm, shown in Figure 29. It consists of a motherboard what is referred to as a carrier board which as a central processing unit integrates an entire computer system with surprisingly large graphics processing capability which is encased within a correspondingly sized card like the same motherboard is.

Figure 29: NVIDIA Jetson AGX Orin system [51].

This system is based on the NVIDIA Jetson AGX Orin that is the card module which consist of the processing system, as mentioned above. The specifications of this system are examined through the individual modules of which it consists as follows:

The Carrier board is the X230D [50] for NVIDIA Jetson AGX Orin, shown in Figure 30. A carrier board provides the essential connections and interfaces to harness the full capabilities of the Jetson module. It acts as a bridge between the Jetson and external devices, simplifying development, prototyping, and integration into diverse systems.

Figure 30: X230D Carrier board [50].

The base specifications for the system are outlined below, providing a comprehensive overview of its key features and capabilities:

- Firstly, the operating voltage is set at 12V, ensuring efficient and reliable performance. The system's Jetson power modes span from 15W to 60W, offering a flexible range of power options to accommodate various tasks and demands.
- To maintain accurate timekeeping, the system incorporates an RTC super cap with a capacity of 200mF. This feature ensures that the system can retain time information even during power interruptions or outages.
- Safety measures are also a priority, with both reverse voltage protection and overvoltage protection in place. These safeguards contribute to the system's longevity and resilience by preventing potential damage from voltage fluctuations.
- In terms of connectivity, the system has an HDMI out port, allowing for seamless external display connections. Additionally, there is a micro-USB 2.0 port and two USB 3.1 ports (Type A), offering versatile options for data transfer and peripheral connections.
- The inclusion of an MCU (Microcontroller Unit) further enhances the system's capabilities, enabling efficient control and coordination of various tasks and components.
- Networking needs are well addressed, with two GbE ports (RJ45) utilizing the RTL8111 PCIe to GbE technology. These ports facilitate high-speed and reliable network connections, crucial for various applications.
- For expansion, the system features PCIe x1 and PCIe x4 slots, denoted as FPC 22 pin (J37) and FPC 40 pin (J20) respectively. These slots allow for additional hardware components to be integrated, enhancing the system's functionalities and adaptability.
- The system's communication capabilities are comprehensive, including both CAN (Controller Area Network) RX/TX and CAN interfaces. Additionally, a UART interface is available, alongside two I2C interfaces for diverse communication needs.
- To cater to imaging and visual data requirements, the system incorporates two CSI-2 interfaces. These interfaces support four lanes each and are equipped with 22-pin connectors, ensuring efficient data transfer and management for imaging applications.

NVIDIA Jetson AGX Orin 32GB Module [51], shown in Figure 31, is an integrated system-on-module powered by the NVIDIA Ampere GPU architecture. This allows it to handle multiple simultaneous AI processes, thanks to its advanced deep learning, vision accelerators, rapid IO, and extensive memory bandwidth. With this card, developers can now tackle intricate AI challenges, ranging from natural language processing to 3D vision and sensor integration. Owing to its compact size, the NVIDIA Jetson AGX Orin 32GB Module is ideally suited for integration into UAVs (drones). This compactness provides powerful onboard AI capabilities without significant weight addition, making it a prime choice for advanced drone applications that demand real-time processing.

Figure 31: NVIDIA Jetson AGX Orin 32GB Module [51].

The module's comprehensive specifications are detailed below, showcasing its remarkable capabilities and cutting-edge components:

- At the heart of this module lies a potent GPU featuring 1792 cores, built upon the NVIDIA Ampere architecture, and bolstered by 56 Tensor Cores. Operating at a frequency of 939MHz, this GPU delivers impressive graphics and computational performance.
- Driving the module's processing prowess is an 8-core Arm® Cortex®-A78AE v8.2 64-bit CPU. This CPU's advanced architecture ensures efficient and responsive processing across a variety of tasks.
- The module is further equipped with two NVDLA (NVIDIA Deep Learning Accelerator) units, elevating its deep learning capabilities. Additionally, a PVA v2 (Vision Accelerator) is integrated, enhancing the module's ability to handle vision-related tasks effectively.
- Memory is of 32GB of 256-bit LPDDR5 RAM, achieving a remarkable data transfer rate of 204.8GB/s. This memory configuration ensures smooth multitasking and rapid data handling.
- For storage, the module incorporates a 64GB eMMC 5.1 storage solution, providing ample space for essential data and applications.
- Video encoding and decoding are handled adeptly, offering the ability to encode in formats such as 4K60 (H.265), 1080p60 (H.265), and decode in formats like 8K30 (H.265) and 1080p60 (H.265), among others. This prowess in video processing makes the module suitable for multimedia-rich applications.
- Camera capabilities are extensive, supporting up to 6 cameras (16 via virtual channels*). With 16 lanes of MIPI CSI-2 D-PHY 2.1 (up to 40Gbps) and C-PHY 2.0 (up to 164Gbps), the module can effectively handle camera inputs for various applications.
- Connectivity options abound, including PCI Express configurations of up to 2 x8, 1 x4, and 2 x1, all supporting PCIe Gen4. USB connectivity comprises 3x USB 3.2 Gen2 (10 Gbps) and 4x USB 2.0 ports, catering to a range of peripheral devices.
- Ethernet connectivity is robust, with 1x GbE and 1x 10GbE interfaces available, ensuring reliable and high-speed network connections.
- The module's visual output is remarkable, offering support for an 8K60 multi-mode DP 1.4a (+MST)/eDP 1.4a/HDMI 2.1 display. This versatility in display options enables seamless integration into various visual setups.
- Diverse I/O options are accessible, including 4x UART, 3x SPI, 4x I2S, 8x I2C, 2x CAN, as well as PWM, DMIC, DSPK, and GPIOs, catering to a wide array of communication and interfacing needs.
- In terms of form, the module has a compact 100mm x 87mm size, featuring a 699-pin Molex Mirror Mezz[11] Connector and an integrated Thermal Transfer Plate, ensuring efficient heat dissipation and mechanical stability.

[11] offer a dense pin field with up to 270 differential pairs in a compact, low-profile hermaphroditic design

Collectively, these specifications paint a picture of a highly capable module, designed to handle intricate computations, advanced AI tasks, multimedia processing, and seamless connectivity across a diverse range of applications.

Storage disk is also included, in the presented system. The disk that is used is the Samsung PM9A1 SSD 1TB M.2 PCI Express 4.0, as shown in Figure 32.

Figure 32: SSD 1TB Storage Disk [69].

Heat sink and a standard fan (80x80mm) are completed the microcomputer. Because of difficult graphical processing the microcomputer is under over-heating and an appropriate cooling system is necessary. This situation requires both a heat sink and a cooler fan. The heat sink covers all the surface of the NVIDIA module. Also, between two surfaces thermal grease is used for better temperature transfer from the module to the heat sink.

3.6.2 Assembly process

In this section is presented every assembly process for the UAV Platform as an inspection tool.

- **Microcomputer:** The following figure shows the assembly process of the system. In this figure, four (4) images can be distinguished, marked with the letters A, B, C and D.

  In the image with the letter A, all the individual parts that make up the microcomputer are enumerated as listed below:

  1. X230D (for NVIDIA Jetson AGX Orin) Carrier board,
  2. Storage disk Samsung PM9A1 SSD 1TB M.2 PCI Express 4.0, as shown in the Figure 33,
  3. NVIDIA Jetson AGX Orin 32GB Module,
  4. Heat sink,
  5. Standard fan (80x80mm),
  6. Protective sieve,
  7. Fan connection cable.

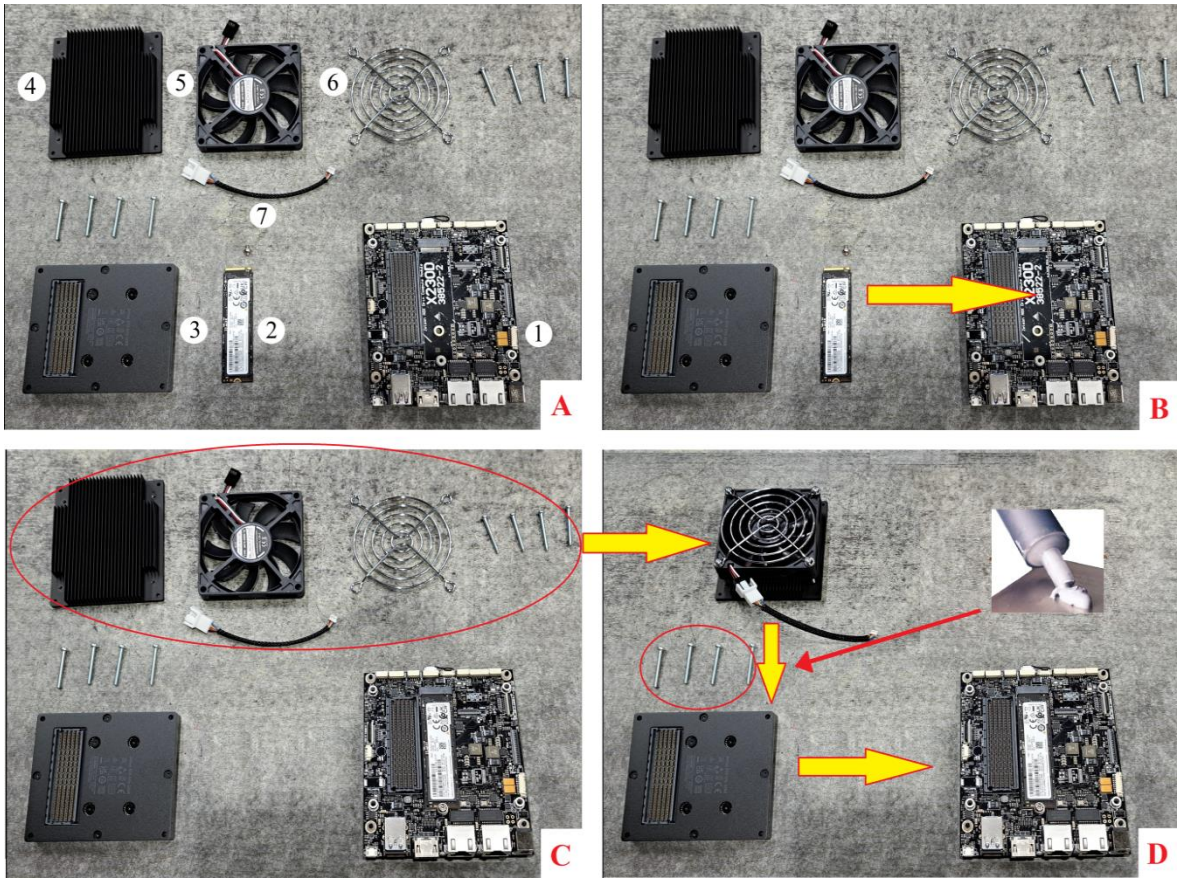

Figure 33: Microcomputer assembly process.

In addition, two sets of four screws each and one small screw above the storage disk appears in the image. Those screws are used to fix the different modules.

In the image with the letter B, the yellow arrow indicates the position where the storage disk is being placed on the carrier board. The disk is placed in a specific slot and the small screw above is used to lock it permanently in position.

In the picture with the letter C, the red ellipse includes those units that make up the cooling system. The yellow arrow indicates the final appearance of cooling system that exists in the next picture with the letter D.

In the finale picture with the letter D, a series of yellow arrows indicates the order in which the modules are placed on top of each other. In the upper right corner of the same picture the thermal grease is shown which is applied onto the surface of the NVIDIA module before the cooling system is mounted. Finally, the NVIDIA module together with the cooling system is placed on the carrier board and locked with the last four screws. The connection cable is connected to the corresponding socket and the microcomputer is ready.

- **Microcomputer mounting on UAV:** The integration of the Jetson companion computer could be held on the outer surface of the drone after careful examination in order to ensure that it aligns with the drone's aerodynamics. This strategic placement is designed to minimize any adverse effects on the UAV's flight characteristics and due to limited workspace in the main hull. By securely affixing the Jetson on the external frame, it avoids interference with the critical airflow patterns that govern the drone's stability and performance. This integration not only preserves the drone's aerodynamic profile but also leverages the Jetson's computational capabilities for tasks like real-time image processing and decision-making without compromising flight efficiency.

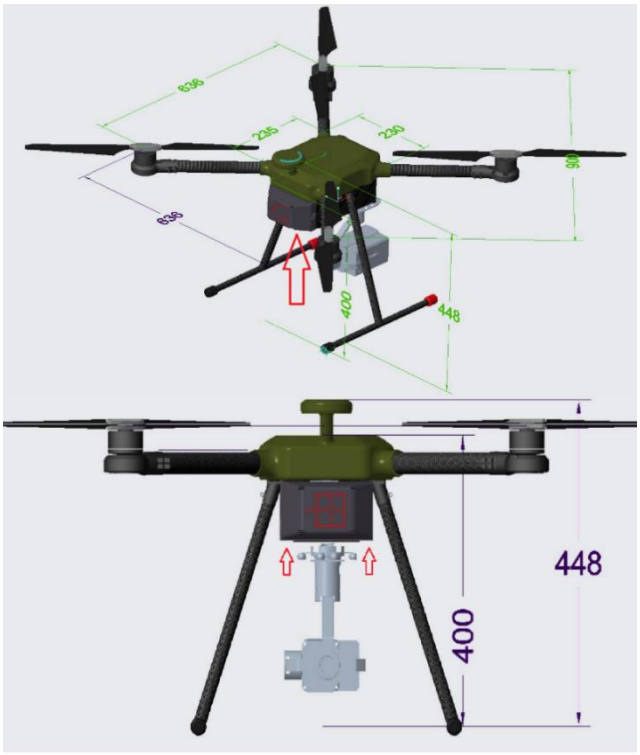

Figure 34: Red Arrows indicate Jetson's position on ATLAS Model.

Also, vibrations in the outer area will be considered. Securing the Jetson companion computer from vibrations is critical to ensure its reliability and functionality within the drone's operational environment. To achieve this, a combination of vibration dampening and mounting techniques is typically employed. The Jetson is often enclosed within a specialized, shock-absorbing casing or foam padding that mitigates vibrations transmitted through the drone's frame. Additionally, carefully engineered mounting brackets or isolators, designed to absorb, and dissipate vibrations, are used to attach the Jetson to the drone's structure. These measures effectively shield the Jetson from the potentially disruptive effects of vibrations caused by the drone's motors and propellers, maintaining the computer's performance.

Figure 35: Example of Jetson mounts to reduce vibration.

3.6.3 Internal Units' Connections and communications

The following Figure 36 shows the connection diagram of the internal modules of the UAV platform (ATLAS 204 N22 [47] based) related to the data transfer from the IP EO/IR camera (Z10TIR) to the microcomputer (NVIDIA Jetson AGX Orin system), as well as the connection of all of them to the communication system that links the inspection tool to the ground control station (GCS).

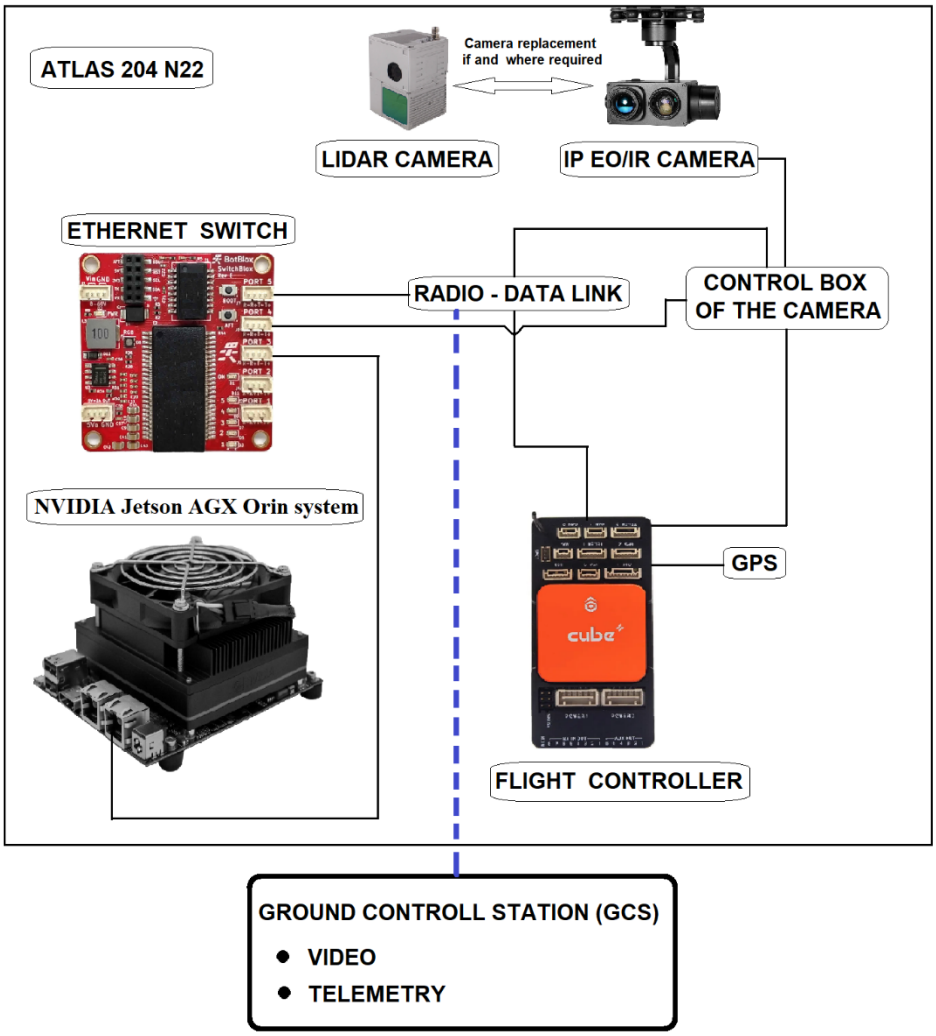

Figure 36: Inspection Tool: “Internal Connections Diagram and Communication”.

The autopilot system (or Flight Controller) serves as the "brain" of the quadcopter, responsible for flight control and navigation. It connects to various sensors, including accelerometers, gyroscopes, GPS and magnetometers, which provide data about the quadcopter's orientation, movement and position. The autopilot also interprets sensor data and sends commands to the motors to adjust the quadcopter's position and attitude. The data that is interpreted from the sensors is called telemetry data and provides useful information about the drone’s state to the user. Telemetry involves the transmission of data, such as GPS coordinates, altitude, speed, and battery status, from the UAV to the ground station in real-time. This bidirectional communication enables operators to monitor the UAV's status, make critical decisions, and adjust flight parameters remotely. Autopilot, on the other hand, being the UAV's autonomous control system, processes telemetry data and executes predefined flight plans or responds to operator commands. It stabilizes the UAV, manages flight dynamics, and ensures safe and precise navigation. Autopilot systems empower UAVs with the ability to conduct a wide range of missions, from aerial photography and mapping to search and rescue, all while maintaining a strong connection to operators on the ground. According to the diagram, the external sensors that are connected to the autopilot’s appropriate ports are: GPS, Radio and Camera Controller/Camera. The GPS module is a crucial component for navigation and location tracking and is typically mounted on top of the quadcopter to ensure a clear view of the sky. The GPS module is connected to the autopilot, providing real-time location data, altitude, and heading information. Next, the radio link is a combination of the airborne unit (receiver) and ground unit (transmitter). The transmitter is integrated in the GCS, used by the operator on the ground, and controls the quadcopter remotely. It

communicates wirelessly with the quadcopter's onboard radio receiver. This connection allows the operator to send control commands to the autopilot, such as adjusting throttle, pitch, roll, and yaw. The receiver relays these commands to the autopilot, enabling manual control or intervention when necessary. Also, telemetry data and video feed are transferred to the ground station due to this radio link connection and bi-directional communication.

Jetson, camera controller and radio receiver are all connected to an Ethernet switch, providing data transfer by combining multiple IP connections. This allows for efficient communication between components and can be essential in scenarios where large data volumes need to be processed in real-time. The camera can provide live video feeds for remote monitoring, object detection, or mapping purposes, then the Jetson processes visual data and can relay relevant information to the autopilot for specific tasks.

3.6.4 Relay Drone System

In the SUNRISE system, in order to control the structure, it is necessary the site of eye to have visual contact between the UAV Platform and its operator. However, the structures are not in easily accessible areas and not in places where the above condition can be fulfilled. For this reason, when such an issue arises it must be solved. There are various ways to incorporate intermediate relay points, such as the installation of fixed ground transmission points, the use of a satellite (Star Link) or the use of a relay drone. Due to the impassability of the area, the use of fixed ground-based relay points is not a feasible solution. Also, Star Link does not consistently provide the necessary data bandwidth required. The solution that works for the SUNRISE system is the relay drone. Figure 37 below shows such a system, and is followed by its description.

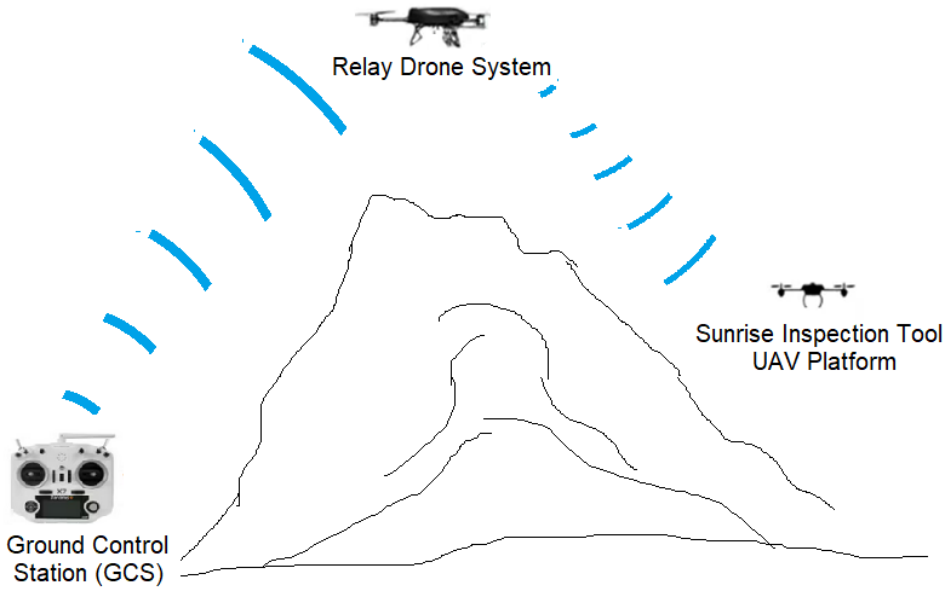

Figure 37: Relay Drone System in operation.

A relay drone system is a network of multiple drones working together to achieve a common goal, with one or more drones serving as relay nodes to extend the communication or operational range of other drones. They achieve this enhancement by functioning as nodes that relay communication between the base station and the operational drone, extending the line of sight (LOS). This involves amplifying the communication signal at each node to compensate for signal power loss due to distance traveled. Additionally, they establish a direct line of sight path between the base station and the operational drone, which further aids in communication.

The network of the project consists of one operational drone and one relay; the latter will transfer video and telemetry information beyond the pilot's line of sight, i.e., remote areas to establish a communication link, extend the operational range of drones, or provide additional sensor coverage. A central ground control station controls the entire relay drone system. The operator can manage the mission, coordinate the relay nodes, and make decisions based on the information received from the data-carrying drones. Relay drone systems often require dynamic coordination as drones move and adapt to changing conditions. The system may employ algorithms and protocols for path planning, task allocation, and re-routing of data as necessary. Once the mission objectives are achieved, the relay drone system may return to the base or continue to operate in a standby mode until needed again.

The relay drone may communicate with the operational drone in real time through the LOS radio and transmit it to the control station.

4 User interface for remote infrastructure inspection

The primary goal of the user interface (UI) for remote infrastructure inspection is to furnish users during the inspection real-time images about areas/components/points of failure of a Critical infrastructure like damaged components, structural issues, corrosions, obstructing vegetation status etc. This is achieved by analyzing image and video feeds and presenting outputs from AI-assisted components into the UI. The design and implementation of the user interface as well as the inspection functionalities provided that required by SUNRISE project, have been based on the following components:

- Two inspection data sources which are a UAV platform and satellite imagery. Both systems through AI technologies can detect the types of problems CI operators are concerned about.
- The interconnection and exploitation of any legacy systems which may at the CI of our interest
- The visualization of the imports in the form of lists of events. These imports are data that have already been annotated by the corresponding inspection tool.
- The reporting services where statistics about the inspections as well as accessing historical information will be provided – if existing.

4.1 General context

As main functionalities, the user interface (UI) tools and dashboards furnish a dynamic web platform, empowering end-users to have a complete engagement with all inspection infrastructures' components of SUNRISE system. This module incorporates contemporary tools and presents a map where during inspection, constantly refreshed with real-time data on inspection point. This feature enhances the inspection process, enabling end-users to accomplish a full range of inspect activities, receive event-driven messages, that means anomalies detection with use of AI algorithms-based methods.

Additionally, data obtained from both inspection sources, (UAV platforms and satellites), referred also as AI-assisted remote inspection tools, has been meticulously annotated to encompass all pertinent inspection-related information derived from the inspection tool.

Finally, the developing user interface and potential legacy CI systems integrations represent the SUNRISE platform's primary interface, catering to in-field inspection even in difficult access point by man. Upon logging into the UI, each platform user with an assigned role is gain access to their application area. The UI operators have been empowered to:

- Visualize data tied to geographical locations on an interactive map. This map can be manipulated using tools like panning and zooming.
- Generate CI maps by incorporating custom points (referred to as Points of Interest or POIs) and areas of significance (known as Areas of Interest or AOIs). These additions can pertain to elements not yet represented on the existing map.
- Overlay various types of data using geo-referenced coordinates such as various topographical data, ground, single and aggregated data (events-alerts), geolocation trails, points, and regions of interest.
- Multimedia content from SUNRISE's subsystems— like images and videos files—can be displayed through the interface.
- Real-time presentation of events and incidents will be showcased in a user-friendly manner.

Overall, the user interface serves as the primary gateway for platform users to engage with monitoring and operational aspects of inspection, facilitating efficient access to infrastructure's critical data and inspection's functionalities.

4.2 Architecture: high level design

The selected architecture on which the development of the UI is based is shown in Figure 38 and is described 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 processes 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.

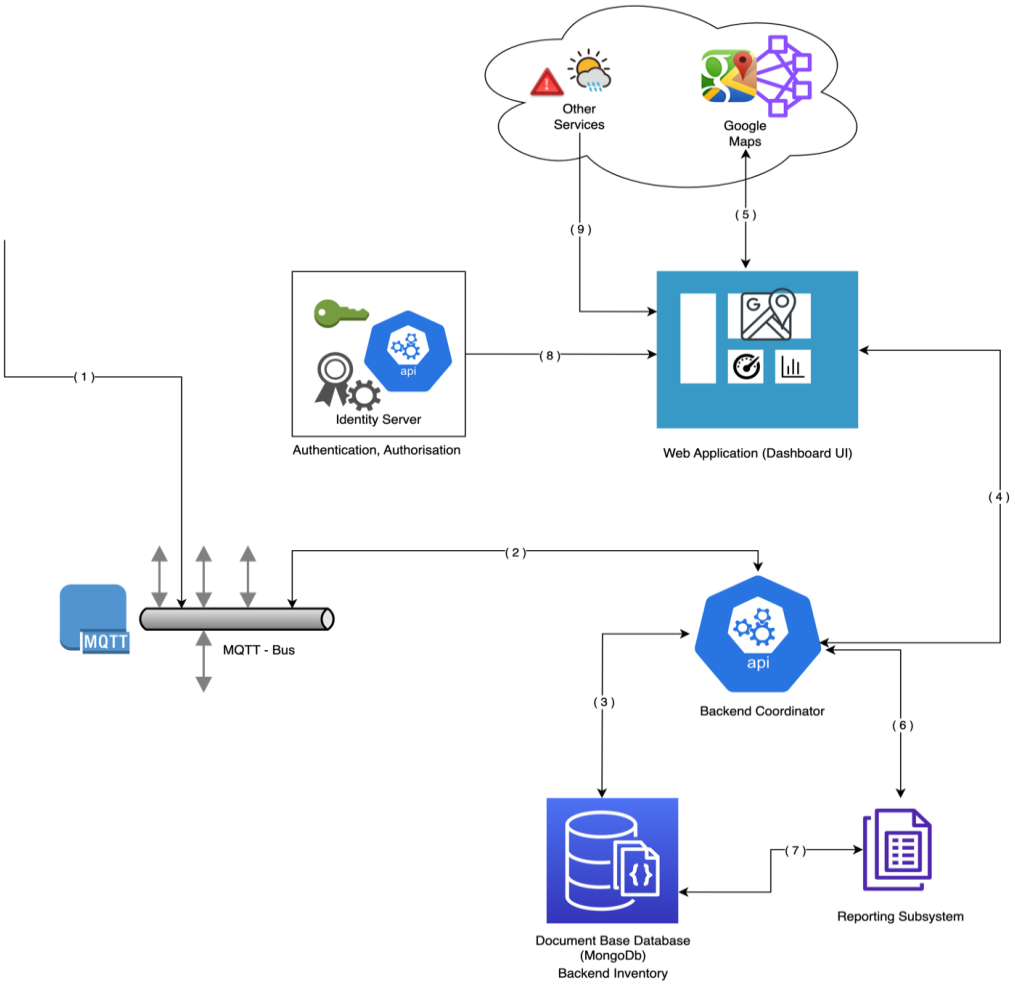

Figure 38: UI Architecture Diagram.

Figure 38, above, illustrates the dashboard UI tool, which operates as a web application. The functionality of this dashboard is underpinned by a robust backend API. This description is elaborated in two distinct subsections: Internal Components and Technical Specifications.

4.2.1 Internal Components

The dashboard UI consists of several internal components, as listed and described below. Each component supports specific system and end-user requirements.

- **Maps and Events (Alert) Management** that utilizes maps and geospatial services to offer observation and orientation capabilities as the basis of understanding the environmental context and situational awareness. It provides a main interface of inspection activities. It presents several information sets in the same visualization space, such as topographic information, sensor placement and direction, alert notifications, and provides a variety of tools and means for interacting with those elements. The user will be able to pan and zoom around the map and interact with it by a set of tools that perform certain actions, such as showing or hiding independent layers that visualize the different distinct types of information as overlays to the map background. It will allow users to show alerts and access their underlying metadata and historical data.

  Map projections will be further enhanced with by external data provided by the Digital Interfaces and Integration of Existing Infrastructures module. Different types of data streams from external sources will be integrated and modelled appropriately to user friendly visualizations. Different data

types will also be presented through different layers enabling the user to select the ones that are useful for the current view and toggle their visibility.

Map data is constantly updated from the UI tools and dashboards backend subsystem and Existing Infrastructures.

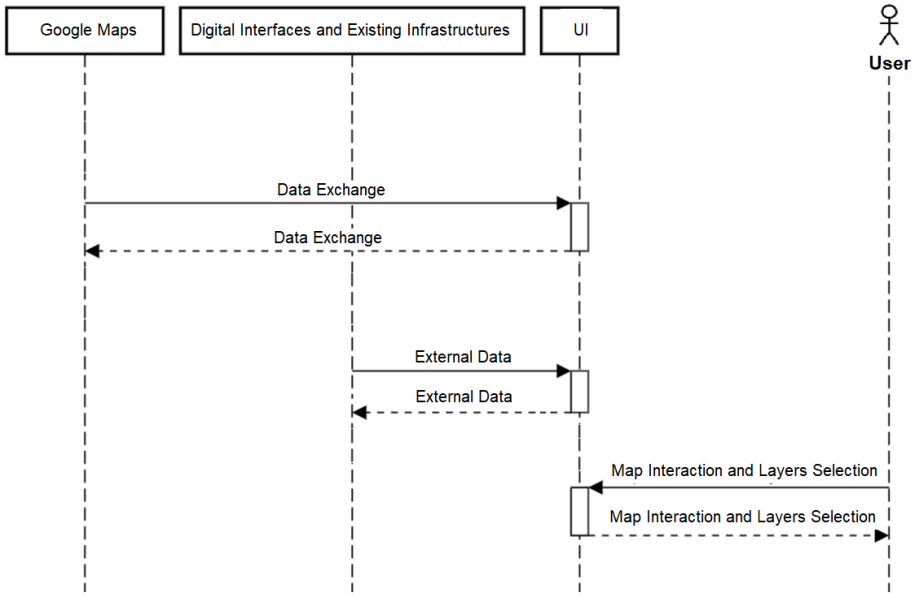

Figure 39: Map Management Sequence Diagram.

As shown in Figure 39 above, the Google Maps and the Digital Interfaces and Existing Infrastructures are updated on the map data constantly. The UI user visualizes the collected data and can tailor the UI to the specific needs and operational conditions, by viewing, examining, and customizing the map (select/save POI/AOI) and selecting data presentation layers.

More specifically, as soon as the AI-assisted remote inspection tools system detects anything, it sends a detection event message on the “events” topic of the MQTT message broker. The backend persists data published, processes it, and publishes the events on the UI subsystem by showing a new Event, through the Event Management sub-component functionality. The Event Management Sequence Diagram is shown in the following Figure 40.

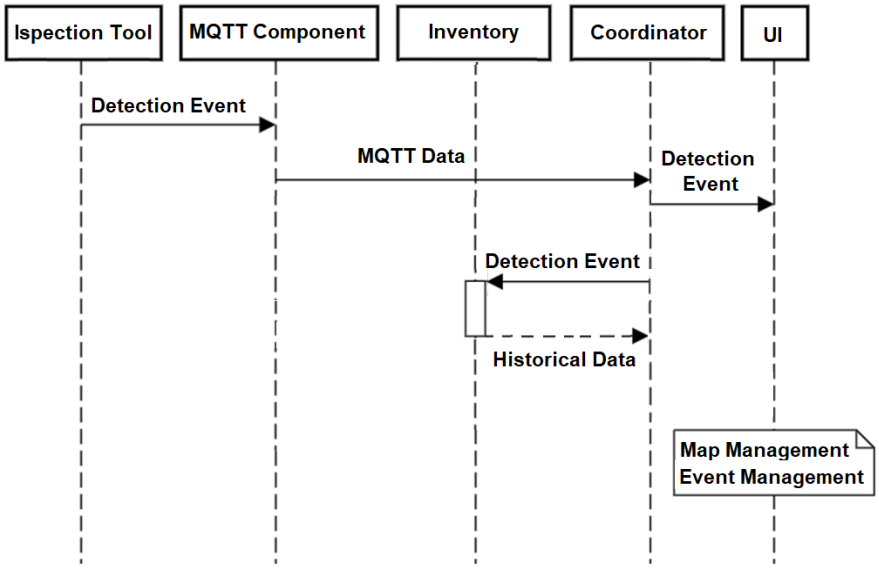

Figure 40: Event Management Sequence Diagram.

The user will be able to access old events from the UI. The user requests to access historical events from the Historical Management sub-component functionality. Backend Coordinator receives the data from Backend Inventory after a historical Data Request to it. Finally, the Historical Management collects the data from the backend Coordinator and presents it in the UI. The Event Management of Historical Data Sequence Diagram is shown in the following Figure 41.

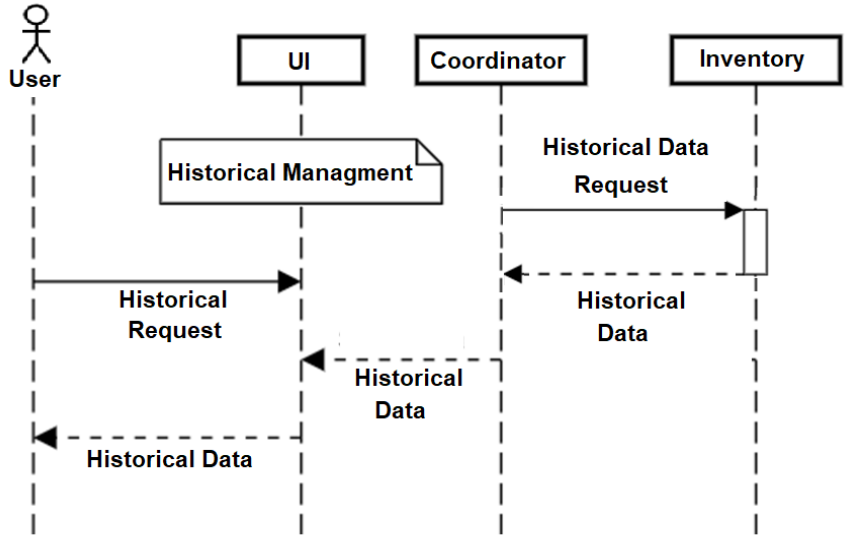

Figure 41: Event Management (Historical Data) Sequence Diagram.

- **Backend Inventory Management** provides management capabilities to the backend subsystem repository/registry of events and metadata and serves as an access point to the subsystem's event-alerting layer. The component will be able to organize all types/categories of event metadata and, through the provided UI, it will enable users to view group types/categories of events and provide access to individual event information such as type of event data, category of events, geospatial data, and other deployment information, stored in the Backend Inventory.
- **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 will be able to access different UI system functions and interfaces. The audit logging mechanism will 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 (Authentication, Authorization and Audit Logging component) leads to the architecture and protocols shown in Figure 42.

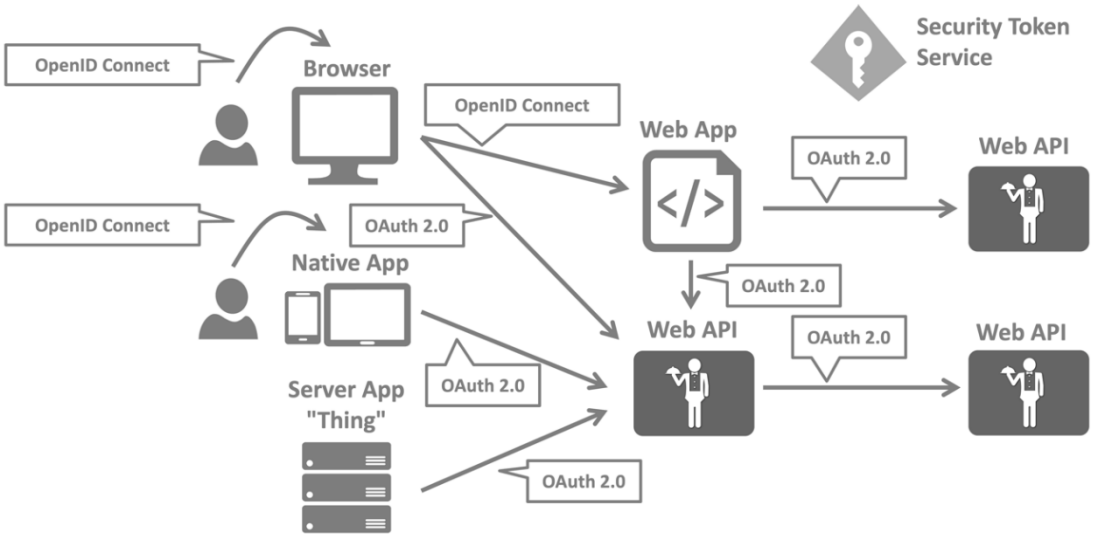

Figure 42: Security Token Architecture and Protocols [70].

The Security Token Service (STS), encompassing Authentication, Authorization, and Audit Logging components, will feature a dedicated administration web application. Through this Administration UI, we will oversee all internal aspects of the service, including Clients, Resources, Scopes, Users, and Roles. The initial interface of the application is depicted in the accompanying Figure 43.

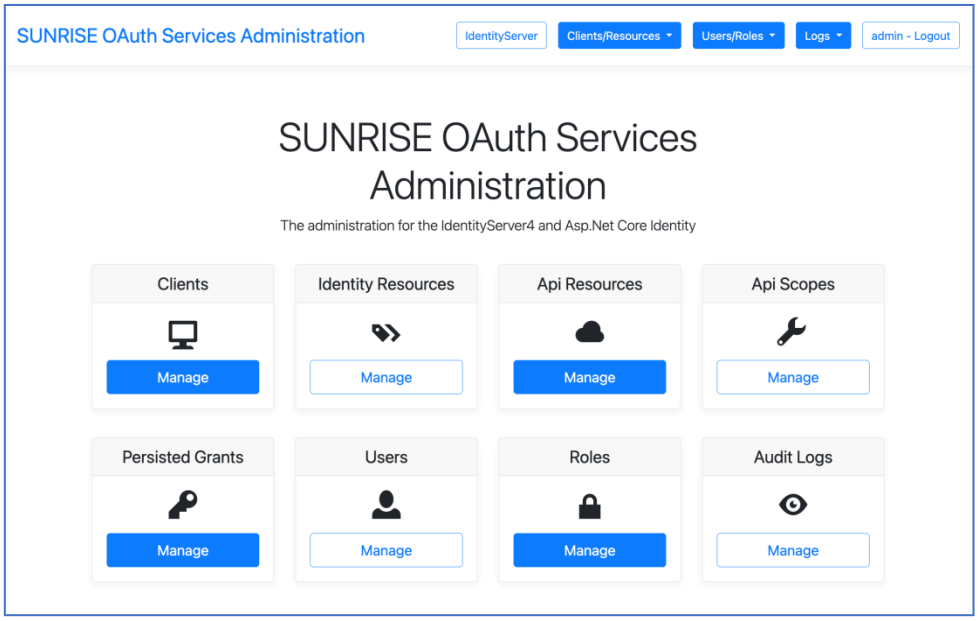

Figure 43: Security Token Service Administration UI.

Among the pivotal functionalities, the Role Management feature stands out prominently. Each role will be linked with specific access rights within the Dashboard UI application. A single role may encompass multiple users, thereby granting access to the Dashboard UI according to the access

rights inherited from the roles to which users belong. The subsequent interface portrays the central screen of the Role Management feature within the Security Token Service (STS).

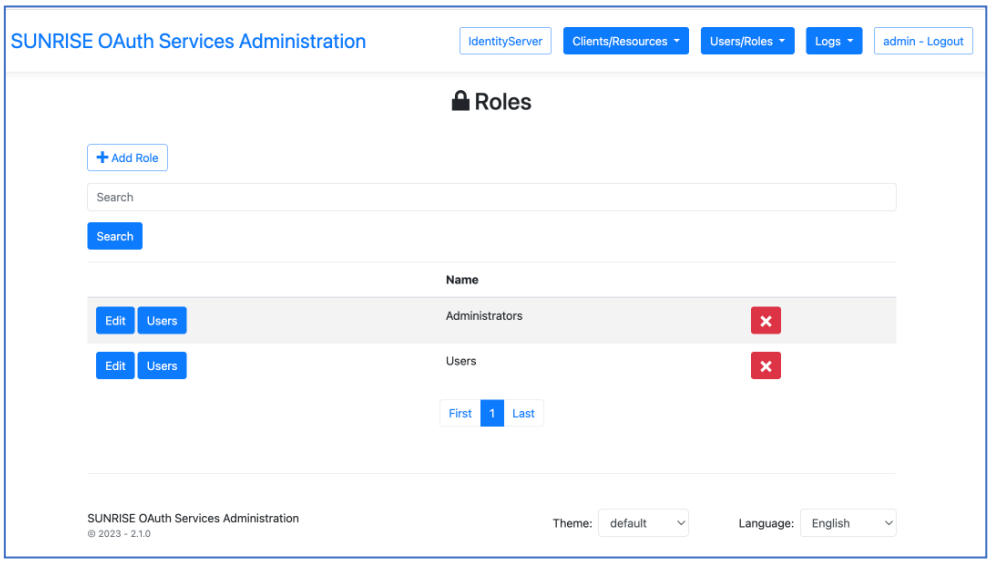

Figure 44: Roles Management UI.

Our approach encompasses the management of the user repository within the Security Token Service (STS) infrastructure. The primary interface of the user management feature will resemble the illustration below.

For each entity management screen, a consistent structure will be maintained. This structure will include an 'Add Entity' button to facilitate entity addition, an input box to enable search functionality, and a list displaying the corresponding entities (in this instance, users). From this list, administrators can select and modify the desired entity (user). This pattern will be consistently applied for the management of various entities, such as clients, resources, scopes, users, roles, and more.

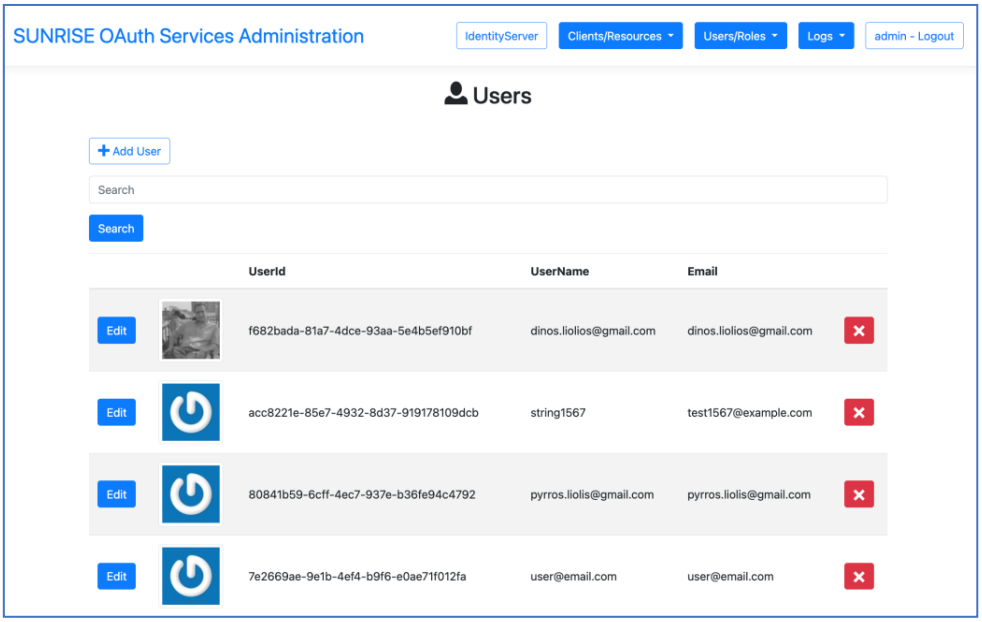

Figure 45: Users Management UI.

Upon selecting the appropriate entity (user), the administrator is directed to the edit entity screen. This interface adopts a classic web form, encompassing all fields pertinent to the entity (user). Each

field employs standard web components, such as an input box for text, an on/off switch, a calendar picker, or a dropdown menu.

If the entity (user) establishes a one-to-many relationship with another entity (e.g., role) administrators can seamlessly navigate to a dedicated web form representing this connection by clicking the corresponding “Manage Entity” button.

Each form is equipped with a 'Save Entity' button, facilitating the preservation of modifications, and enabling a return to the previous screen. These functionalities are visually represented in the accompanying Figure 46.

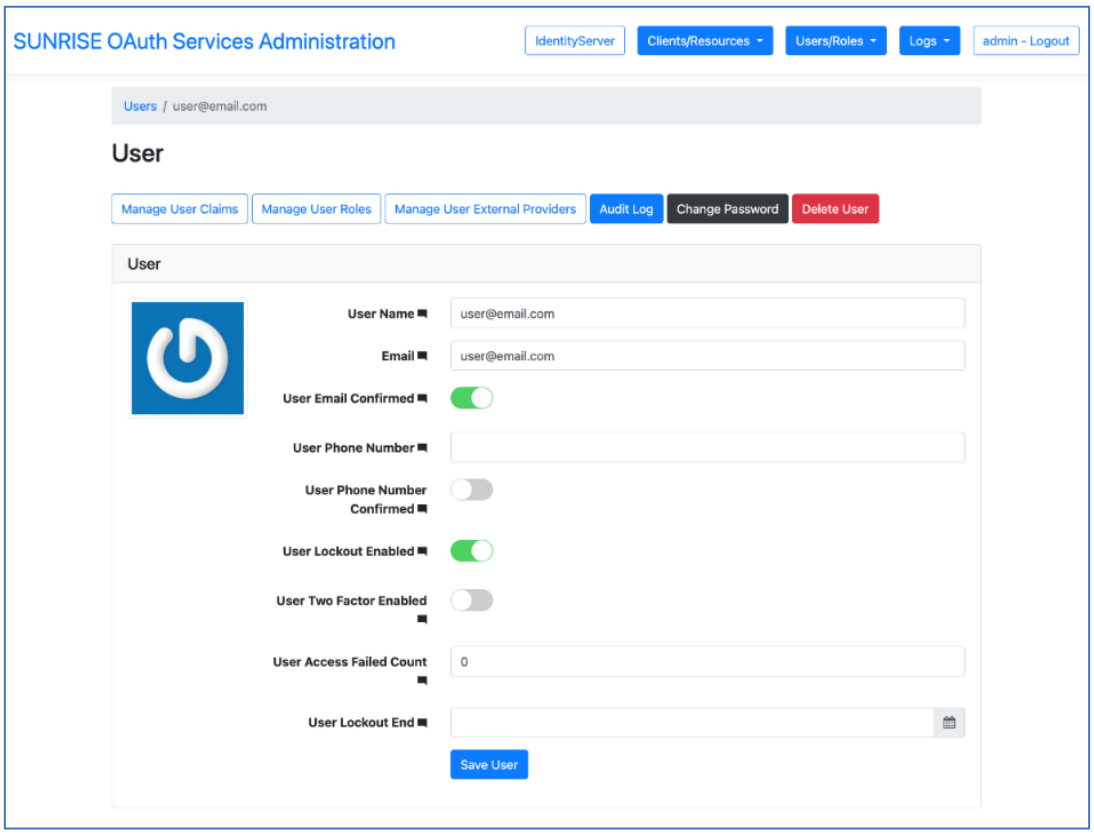

Figure 46: Edit User UI.

- **Reporting** component responsible for preparing and presenting statistical information from data stored in the Backend layer and for providing predefined reports on request. Data for predefined reports will be prepared through processing and aggregation of collected system and user activity data, and presented using appropriate, predefined visualizations (e.g., Tables, Line charts, Bar charts, Pie charts, Heat maps). End-users can export displayed reports in several formats (csv, xlsx, pdf, etc).
- **Video Streaming** responsible for providing flexible and scalable video surveillance capabilities of remote video, audio, and event streams from the AI-assisted remote inspection tools. Multimedia content must be encoded in a suitable format to be displayed in a web browser.
- **Backend Coordinator** serving as a middleware component that manages and orchestrates the streams of geo-data flows between the UI and the Backend Inventory Management, such as event handling and presentation. It will subscribe to the “events” MQTT topic. As soon as the AI-assisted remote inspection tools published a detection event message it will process the event and upload it to the Backend Inventory Management. As soon as the information is ready it will inform the UI that the event is ready for visualization.

4.2.2 Technical Specifications

The technology stack consists of:

- **Angular** (often referred to as "Angular 2+" or simply "Angular") is a modern front-end web application platform developed by Google. It is the successor to AngularJS (Angular 1.x) and was rewritten from the ground up to address the limitations and challenges of its predecessor. Angular offers a comprehensive set of tools and features for building dynamic, single-page web applications (SPAs) and progressive web apps (PWAs).
- **ASP.NET Core** is a free and open-source web framework developed by Microsoft for building modern, cloud-based web applications. It is the next generation of ASP.NET, and was first released in 2016.

  ASP.NET Core provides a modular architecture that allows developers to build web applications using a variety of languages, including C#, F#, and Visual Basic. It is cross-platform and can run on Windows, Linux, and macOS.
  ASP.NET Core includes several key components, including the MVC framework for building web applications, the Razor templating engine for creating views, and the Entity Framework for working with databases. It also includes support for modern web development technologies like Web API, SignalR[12], and WebSockets.[13]
- **MongoDB** is a widely used open-source, NoSQL (non-relational) database management system, as described in [52]. It is designed to store and manage large volumes of data, especially unstructured or semi-structured data, in a flexible and scalable manner. MongoDB diverges from traditional relational databases by using a document-oriented data model instead of tables with fixed schemas. This allows developers to work with data more dynamically and adapt to changing data structures.

  The Sunrise system will use the MQTT messaging protocol to handle JSON object type events. In this case no adaptation system between MQTT and MongoDB is required, because this DB stores data of this type.
  MongoDB is commonly used in various applications, including web and mobile applications, content management systems, data analytics platforms, and more. Its flexibility and scalability make it suitable for scenarios where data structures are dynamic and need to accommodate growth. However, while MongoDB offers benefits, it is important to consider factors like data modeling, indexing strategies, and data consistency based on the specific requirements of your project.
- **Google Maps** JavaScript API is a free, web-based mapping service provided by Google. It allows developers to embed maps, geolocation, and other location-based features into their web applications using JavaScript. The API provides a number of tools and services for building custom maps, markers, and other location-based features.

  The Google Maps JavaScript API is widely used in web development, particularly for location-based applications such as delivery services, ride-sharing apps, and real estate listings. It is free to use for most applications but does have usage limits and pricing for high-traffic applications.
- **MQTT** (Message Queuing Telemetry Transport) is a lightweight messaging protocol designed for IoT (Internet of Things) devices and applications that require efficient, real-time communication between devices and servers. It was first developed in 1999 by Andy Stanford-Clark of IBM and Arlen Nipper of Eurotech and has since become a widely adopted protocol for IoT devices.

[12] WebSockets is an abstraction over some of the transports that are required to do real-time work between client and server.
[13] SignalR is a computer communications protocol, providing full-duplex communication channels over a single TCP connection.

MQTT uses a publish-subscribe messaging model, in which devices can publish messages to a central server or broker, and other devices can subscribe to receive those messages. This model allows for efficient communication and reduces the need for constant polling and data transmission.
MQTT is commonly used in IoT applications for a variety of purposes, including sensor data collection, device control, and monitoring. It is supported by many IoT platforms and frameworks and is considered a key technology in the development of the IoT ecosystem.

4.3 Dashboards mockups

The SUNIRISE login page presents a secure gateway to access the platform. Users input their credentials – a unique combination of username and password – to verify their identity. The page's design is intuitive, with fields for entering credentials prominently displayed. Once verified, users gain authorized entry, unlocking the platform's features and personalized /CI related content. In case of forgotten credentials, the page also provides options for password recovery or account assistance.

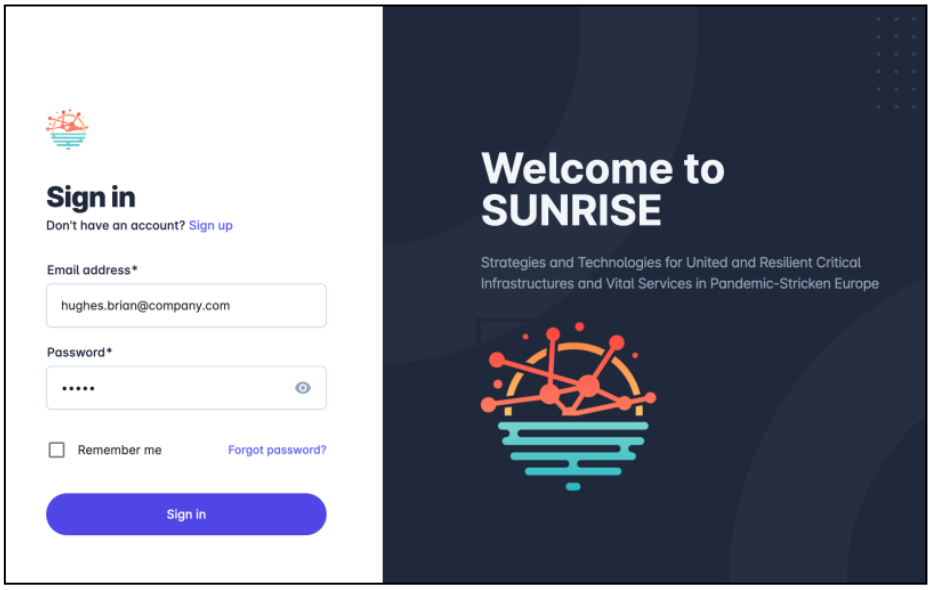

Figure 47: SUNIRISE login page.

After granting access to the SUNRISE platform, the user navigates to the first page with the GIS Map and relevant layers of the infrastructure of their responsibility. On the right side a list of all the events that have been detected is presented with some fundamental information regarding each specific event.

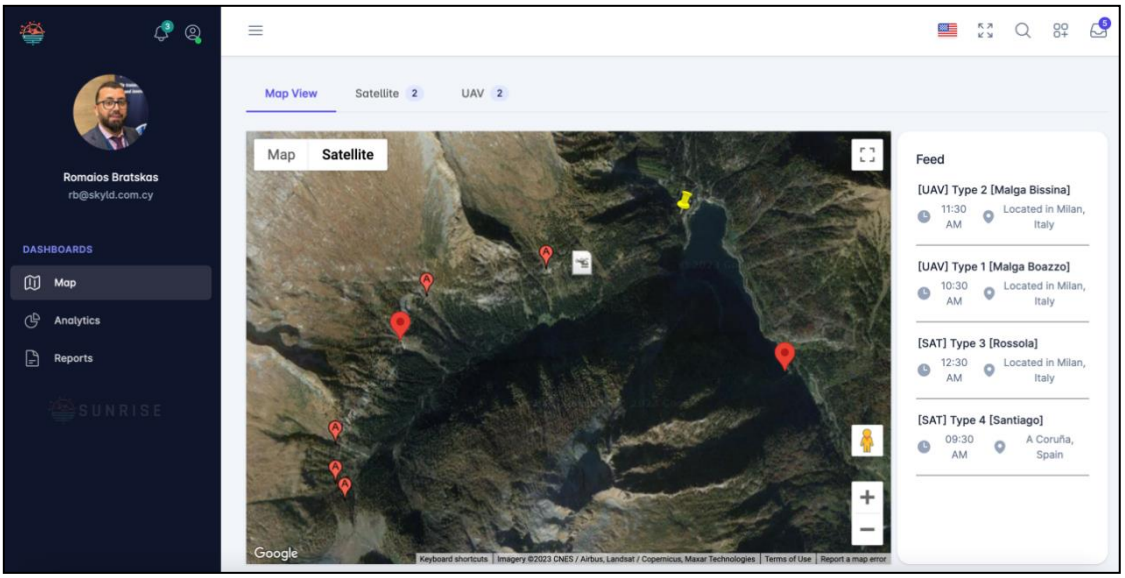

Figure 48: first page with the GIS Map and relevant layers.

By clicking the marker on the map or an event on the Events List on the right side, the annotated images from the inspected infrastructure are shown on a pop-up window in the main screen of the GIS. This pop-up window also presents the fundamental info such as event type, time stamp, location, source of inspection etc.

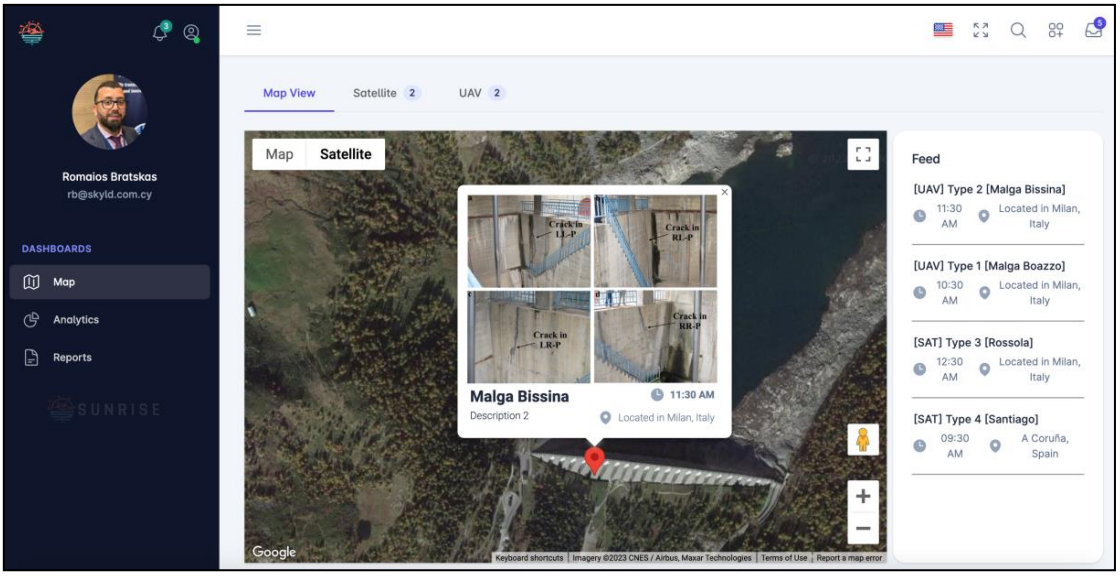

Figure 49: Event Presentation (A).

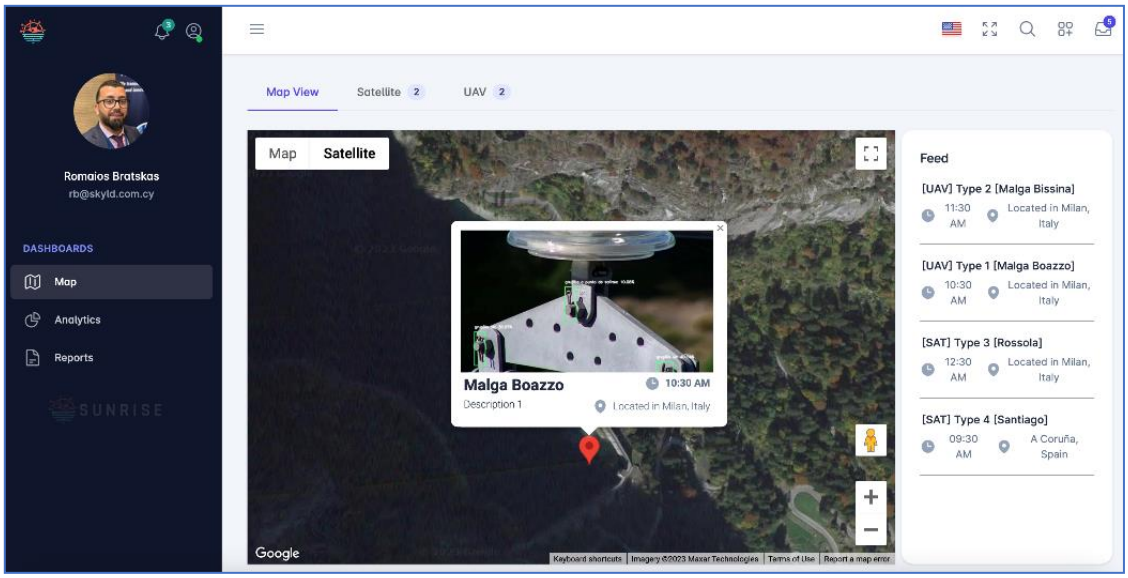

Figure 50: Event Presentation (B).

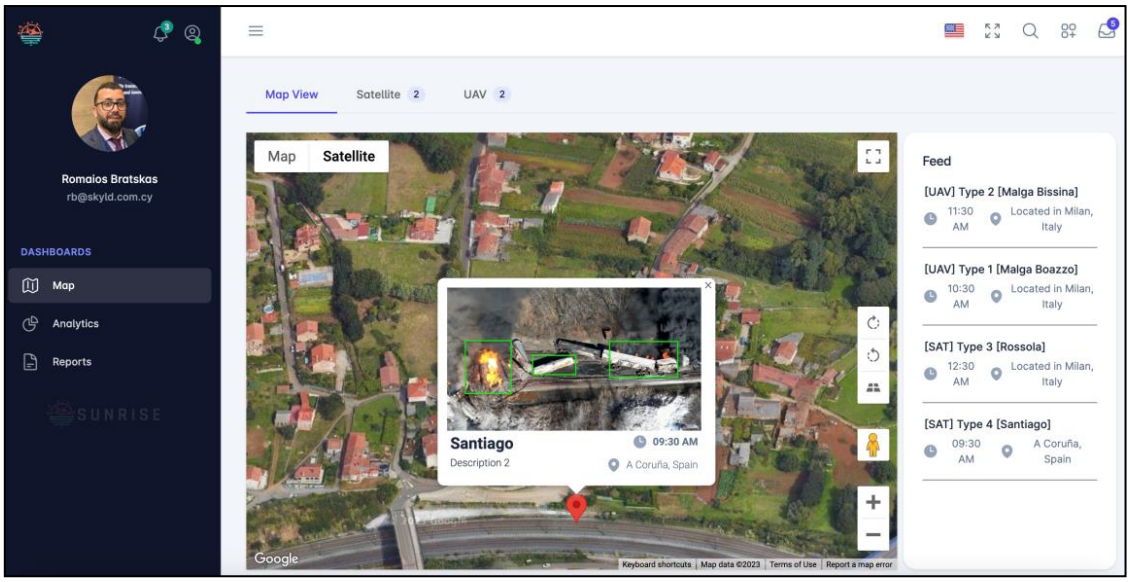

Figure 51: Event Presentation (C).

At the upper part of the map a ribbon with distinct categories is shown. For example, we can have the list of the events by source category (Satellite – UAV).

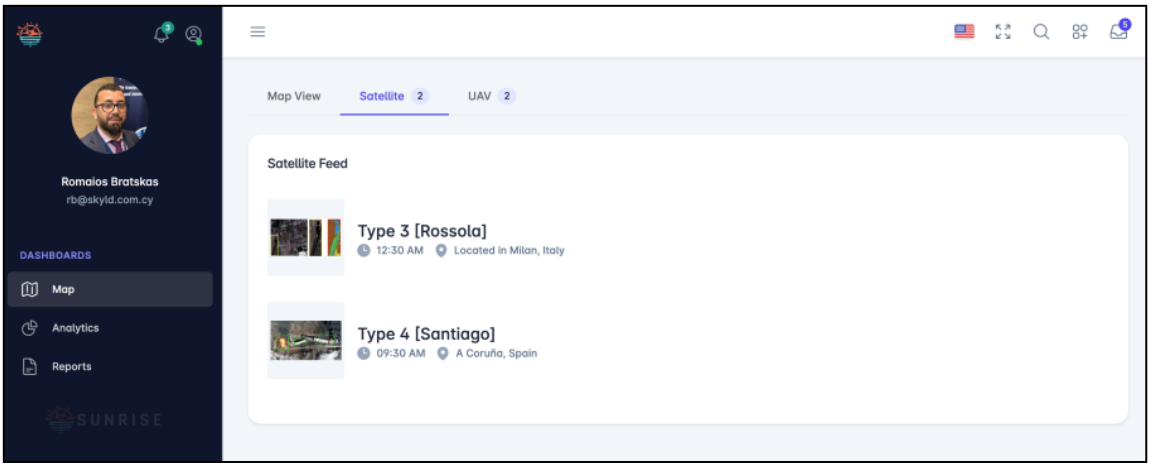

Figure 52: list of the events by source category (Satellite).

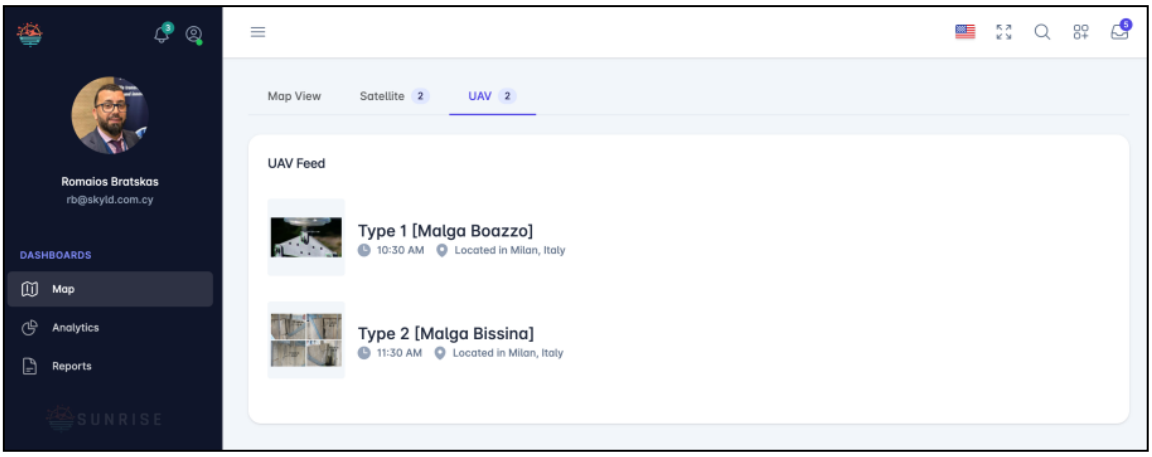

Figure 53: list of the events by source category (UAV).

On both the top ribbon and the vertical ribbon situated on the left side, users can access analytics tools. These tools enable them to select from a range of graphical representations to visualize analytics data on a monthly, semesterly, or yearly basis. Users can utilize these graphics to depict correlations between events detected via UAV or satellite observations. Additionally, they can categorize these events based on types such as corrosion events, leakages, obstructive vegetation, and more.

Figure 54: Analytics View (A).

Figure 55: Analytics View (B).

4.4 Integration and validation

Within SUNRISE, an MQTT system with a central MQTT broker and a publish-subscribe mechanism serves as the integration pathway for ingesting data from three distinct inspection data sources. These sources include the UAV platform, satellite, and potentially any existing legacy systems within the infrastructure of interest. If integrating a legacy system, an adaptation mechanism may be required to modify its outgoing data type accordingly.

The three data sources, as previously mentioned, act as publishers, generating messages containing the data they intend to share with subscribers. The sole subscriber in this scenario is the Backend Coordinator of the dashboard system, which seeks to receive the data. Typically, a subscriber conveys its interest by issuing subscription requests to the MQTT broker.

The MQTT broker serves as a central intermediary that enables seamless communication between publishers and subscribers. Within the SUNRISE system, this broker receives messages from the inspection data sources and effectively transmits them to the dashboard system.

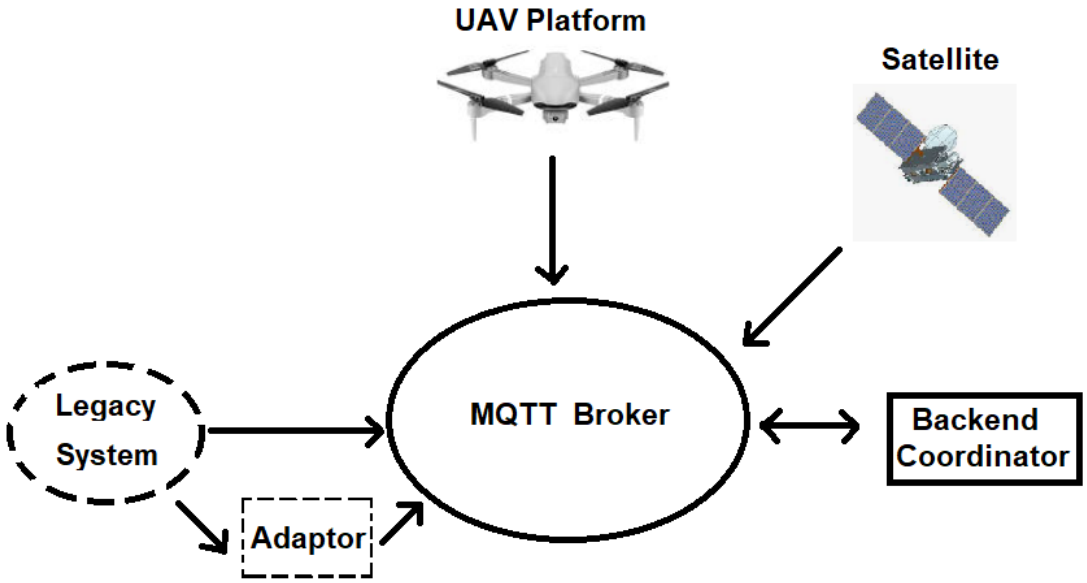

Figure 56: Integration Diagram.

The Integration Process described as follow: The Publishers establish a connection to the MQTT broker and authenticate themselves using credentials. Once connected, publishers can start sending their messages. The MQTT broker receives these Messages containing data and topic information and temporarily stores them. Subsequently, the MQTT broker routes the messages to subscriber based on subscriber's interest. On data integration the subscriber receives the messages that containing relevant data and proceeds as needed. Integration can involve storing data in databases, triggering actions, generating notifications, or updating visualizations.

On Data validation in an MQTT system, a crucial step is to ensure that the incoming data from publishers is accurate, consistent, and conforms to the expected format. This helps prevent erroneous data from being distributed to subscribers and ensures the overall integrity of the system. Data validation in an MQTT system is based on several parameters which are Payload Format Validation, Subject Validation, Data Range and Constraints, Message Size Validation, Protocol Validation, Quality of Service (QoS) and Identification, Flag Validation Preservation, Security Checks and Error Handling.

4.5 Deployment

The UI will be deployed in a Virtual Machine system and will be deployed in a Cloud. All internal components will be deployed in the same way. The operating system that is selected for the SUNRISE dashboard system is a Linux system.

There are several reasons why Linux systems are preferred: A Linux system is an open-source operating system, provides a high level of customization as well as stability and reliability. Also, a Linux is inherently more secure due to its design and the open-source community's active contributions to identifying and fixing vulnerabilities. Regular updates, robust permission models, and security features like SELinux contribute to its security reputation. The Linux supports a wide range of hardware architectures and devices, is highly efficient and optimized for performance, has a vast and active community, is cost-effective as there are no licensing fees associated with using it, offers a rich ecosystem of open-source software applications, tools, libraries, and development frameworks as well as Cloud Compatibility.

Particularly for the deployment of the SUNRISE UI we will use the Ubuntu 22.04.3 LTS that is the latest LTS version of Ubuntu. LTS stands for long-term support, which means five years of free security and maintenance updates, guaranteed until April 2027.

5 Pilot trials execution

The results from the "Pilot 0 - Lab Validation" have been showcased in three demonstrative sessions to the Critical Infrastructure (CI) stakeholders, each corresponding to the primary sections of this document (Sections 2, 3, and 4). The overall feedback received has been favorable, and the insights gathered from the comments have been instrumental in further refining the proposed solutions.

The execution of "Pilot 1" and "Pilot 2" is scheduled for the second and third years of the project, respectively. Detailed descriptions of these pilots can be found in the deliverable D7.1. Since the drafting of D7.1, there have been no significant setbacks or changes to the presented plans, and thus, all proposed scenarios are still deemed feasible.

One notable point to highlight is the observation that, in certain scenarios, particularly at the HDE facilities, it might be challenging to maintain direct visual contact with the UAV due to the terrain's topography. To address this control and communication potential issue, the UAV relay system, as described in Section 3.6, have been proposed.

6 Conclusions

This document, as the first technical implementation outcome from SUNRISE WP7, has provided a comprehensive overview of the tools and methodologies developed for the remote inspection of critical infrastructures. D7.2 will serve as a foundational input for the subsequent D7.3 and other WP7 deliverables, setting the stage for an iterative and evolutionary process.

Firstly, the satellite footage inspection tool and the UAV footage inspection tool offer a high-level architectural design that is both robust and scalable. The modular approach ensures that each component of the tools can be refined independently, allowing for flexibility and adaptability. The laboratory validation of both modules indicates a successful implementation, with the deployment strategies ensuring a wide range of operational contexts can be addressed.

Beyond the positive laboratory results validating the models, it has been showcased how the substantial advancements in computer vision and text analysis fields can be implemented and transitioned into wide-ranging real-world solutions. This aids in narrowing the gap between academic developments and industrial products.

Secondly, the laboratory integration of the UAV platform lays the foundation for an optimal combination of hardware and software components to ensure efficient and effective inspections.

Lastly, the development of the user interface emphasizes user-centric design. The mockups and high-level architectural designs ensure the interface is intuitive and user-friendly. The integration and validation processes underscore the interface's reliability, ensuring users can effectively interact with the satellite and UAV footage inspection tools. As the integration of analysis tools with graphical user interfaces progresses further, the task of establishing more detailed user guidelines will be addressed in subsequent deliverables.

In summary, the tools and strategies presented in this document represent a significant step forward in the remote inspection of critical infrastructures, having achieved the goal of implementing PoCs with a TRL5 or higher at this project stage. The modular designs, combined with state-of-the-art technologies and user-centric interfaces, ensure that the SUNRISE project is well-positioned to address the challenges of inspecting critical infrastructures in a variety of contexts. As the project progresses, it will be essential to continue refining these tools based on feedback from stakeholders and real-world testing scenarios.

References

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

[2] ESA science toolbox exploitation platform, Sen2cor, https://step.esa.int/main/snap-supported-plugins/sen2cor/, retrieved 2023-08-03

[3] Lang N., Schindler, K. & Wegner, J. D. (2019), Country-wide high-resolution vegetation height mapping with Sentinel-2*, Remote Sensing of Environment,* 223:111347.

[4] 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.

[5] 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.

[6] 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.

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

[8] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., ... & Rabinovich, A. (2015), Going deeper with convolutions, Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 1-9.

[9] Chollet, François. Xception: Deep learning with depthwise separable convolutions, Proceedings of the IEEE conference on computer vision and pattern recognition, 2017.

[10]Xiao, T., Liu, Y., Zhou, B., Jiang, Y., & Sun, J. (2018). Unified perceptual parsing for scene understanding, Proceedings of the European conference on computer vision, pp. 418-434.

[11]Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017), Attention is all you need, Advances in neural information processing systems, 30.

[12]Loshchilov, Ilya, and Frank Hutter, "Decoupled Weight Decay Regularization" (2018), International Conference on Learning Representations.

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

[14]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.

[15]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).

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

[17]Bergmann, P., Fauser, M., Sattlegger, D., & Steger, C. (2019), MVTec AD--A comprehensive real-world dataset for unsupervised anomaly detection, *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9592-9600.

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

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

[20]M Rustowicz, Rose, et al. (2019), Semantic segmentation of crop type in Africa: A novel dataset and analysis of deep learning methods*, Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*. pp. 75-82.

[21]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.

[22]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.

[23]Pesaresi, M., Corbane, C., Julea, A., Florczyk, A. J., Syrris, V., & Soille, P. (2016), Assessment of the added-value of Sentinel-2 for detecting built-up areas, *Remote Sensing*, 8(4), 299.

[24]Sibanda, M., Mutanga, O., & Rouget, M. (2015), Examining the potential of Sentinel-2 MSI spectral resolution in quantifying above ground biomass across different fertilizer treatments, *ISPRS Journal of Photogrammetry and Remote Sensing*, 110, 55-65.

[25]Otunga, C., Odindi, J., Mutanga, O., & Adjorlolo, C. (2019), Evaluating the potential of the red edge channel for C3 (Festuca spp.) grass discrimination using Sentinel-2 and Rapid Eye satellite image data, *Geocarto International*, 34(10), 1123-1143.

[26]Bruzzone, L., Bovolo, F., Paris, C., Solano-Correa, Y. T., Zanetti, M., & Fernández-Prieto, D. (2017), Analysis of multitemporal Sentinel-2 images in the framework of the ESA Scientific Exploitation of Operational Missions, *9th International Workshop on the Analysis of Multitemporal Remote Sensing Images (MultiTemp)*, pp. 1-4.

[27]Sitokonstantinou, V., Papoutsis, I., Kontoes, C., Lafarga Arnal, A., Armesto Andrés, A. P., & Garraza Zurbano, J. A. (2018), *Scalable parcel-based crop identification scheme using Sentinel-2 data time-series for the monitoring of the common agricultural policy,* Remote Sensing, 10(6), 911.

[28]GeeksforGeeks (n.d.) Singleton Design Pattern. Available at: https://www.geeksforgeeks.org/singleton-design-pattern/ (Accessed: 21 September 2023).

[29]Touvron, H., Martin, L., Stone, K., & et al. (2023). Llama 2: Open Foundation and Fine-Tuned Chat Models. GenAI, Meta. Available at: https://arxiv.org/pdf/2307.09288.pdf

[30]OpenAI (2023). GPT-4 Technical Report. Available at: https://arxiv.org/pdf/2303.08774.pdf

[31]Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G. & Sutskever, I., 2021. Learning Transferable Visual Models From Natural Language Supervision. arXiv:2103.00020v4 [cs.CV], [online] Available at: https://arxiv.org/pdf/2103.00020.pdf.

[32]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.

[33] Rombach, R., Blattmann, A., Lorenz, D., Ommer, B. & Esser, P., 2022. High-Resolution Image Synthesis with Latent Diffusion Models. Ludwig Maximilian University of Munich & IWR, Heidelberg University, Germany. Available at: https://arxiv.org/pdf/2112.10752.pdf.

[34] 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.

[35] 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.

[36] COCO Consortium, 2023. Common Objects in Context (COCO) Dataset. [online] Available at: https://cocodataset.org [Accessed 6 September 2023].

[37] 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.

[38] Kirillov, A., Mintun, E., Ravi, N., Xiao, T., Whitehead, S., Berg, A.C., Mao, H., Rolland, C., Gustafson, L., Lo, W-Y., Dollár, P. & Girshick, R., 2023. Segment Anything. Meta AI Research, FAIR. Available at: https://arxiv.org/pdf/2304.02643.pdf.

[39] 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.

[40] Ultralytics, 2023. Ultralytics GitHub Repository. [online] Available at: https://github.com/ultralytics/ultralytics [Accessed 6 September 2023].

[41] Terven, J.R. & Cordova-Esparza, D.M., 2023. A Comprehensive Review of YOLO: From YOLOv1 and Beyond. CICATA-Qro, Instituto Politecnico Nacional & Facultad de Informática, Universidad Autónoma de Querétaro, Mexico. Available at: https://arxiv.org/pdf/2304.00501.pdf.

[42] 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.

[43] Alayrac, J-B., Barr, I., Hasson, Y., Donahue, J., Luc, P., Miech, A., Lenc, K., Mensch, A., Millican, K., Reynolds, M., Ring, R., Rutherford, E., Gong, Z., Samangooei, S., Borgeaud, S., Brock, A., Binkowski, M., Barreira, R., Cabi, S., Monteiro, M., Han, T., Menick, J., Nematzadeh, A., Sharifzadeh, S., Vinyals, O., Zisserman, A. & Simonyan, K., 2023. Flamingo: a Visual Language Model for Few-Shot Learning. DeepMind. Available at: https://arxiv.org/pdf/2301.12597.pdf.

[44] Goyal, Y., Khot, T., Summers-Stay, D., Batra, D. & Parikh, D., 2017. Making the V in VQA Matter: Elevating the Role of Image Understanding in Visual Question Answering. Virginia Tech, Army Research Laboratory & Georgia Institute of Technology. Available at: https://arxiv.org/pdf/1612.00837v3.pdf.

[45] 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.

[46] 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.

[47]ATLAS 204. Available at: https://altus-lsa.com/wp-content/uploads/2022/07/ATLAS-204-BROCHURE-ENG-.pdf

[48]Viewpro (n.d.) 'Article details Z10TIR infrared and visual spectrum camera'. Available at: http://www.viewprotech.com/index.php?ac=article&at=read&did=206 (Accessed: 20 September 2023).

[49]GeoSun Lidar (n.d.) 'GeoSun Gairhawk Sesries GS-100C Lidar Scanning System Entry Level 3D Data Collection Livox Avia Sensor'. Available at: https://www.geosunlidar.com/sale-13560662-geosun-gairhawk-sesries-gs-100c-lidar-scanning-system-entry-level-3d-data-collection-livox-avia-sens.html (Accessed: 20 September 2023).

[50]Auvidea (n.d.) 'Product: 70417'. Available at: https://auvidea.eu/product/70417/ (Accessed: 20 September 2023).

[51]NVIDIA (2022) 'Jetson AGX Orin 32GB Module Now Available'. Available at: https://developer.nvidia.com/blog/jetson-agx-orin-32gb-module-now-available (Accessed: 20 September 2023).

[52]Kabir, A. (12 August 2023) 'What is MongoDB?'. Available at: https://medium.com/@asifkabiremon/what-is-mongodb-a1c1572426f2 (Accessed: 20 September 2023).

[53]Planet Fusion Monitoring, Planet Labs, https://assets.planet.com/docs/Planet_fusion_specification_March_2021.pdf, retrieved 25.09.2023

[54]Sentinel-2, ESA, https://www.esa.int/Applications/Observing_the_Earth/Copernicus/Sentinel-2, retrieved 25.09.2023

[55]Chen, L. C., Papandreou, G., Kokkinos, I., Murphy, K., & Yuille, A. L. (2017). Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. IEEE transactions on pattern analysis and machine intelligence, 40(4), 834-848.

[56]Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 580-587).

[57]Domingos, P. (2012). A few useful things to know about machine learning. Communications of the ACM, 55(10), 78-87.

[58]https://news.agu.org/files/2021/10/landslide_timo-volz-unsplash.jpg

[59]https://www.mangaloretoday.com/contentfiles/2020/jul/Kulshekar16jul2021.JPG

[60]https://www.climatechangepost.com/media/news/2019/08/Flooded_railway_tracks_England_2007.jpg.820x520_q95_crop-smart.jpg

[61]https://www.pexels.com/video/drone-footage-of-an-aqueduct-6943203/

[62]https://www.arabnews.com/sites/default/files/styles/n_670_395/public/2015/06/27/file-27-1435393414619607900.jpg?itok=e-v-BPve

[63]https://www.checkatrade.com/blog/wp-content/uploads/2020/06/water-supply-pipe-repair-cost.jpg

[64]https://www.cprplumbingservices.com/wp-content/uploads/2018/07/water-pipe-leak.jpg

[65]https://www.fladerplumbing.com/wp-content/uploads/2021/12/iStock-1319279780Jan-1024x768.jpg

[66]https://electricknowhow.com/wp-content/uploads/2022/09/image-19.png

[67]https://149358052.v2.pressablecdn.com/wp-content/uploads/2017/11/Pg-104-678x381.jpg

[68]https://www.shutterstock.com/es/image-photo/power-line-insulator-broken-replaced-high-2313270247

[69]https://cdn3.volusion.com/kvxpe.qrwzs/v/vspfiles/photos/MZVL21T0HCLR-00B00-2T.jpg?v-cache=1695627843

[70]https://cdn.hashnode.com/res/hashnode/image/upload/v1674163307044/38978829-b2dd-46e1-8a3e-1d25a874cc93.png?auto=compress,format&format=webp