

EYE-Sense: Empowering Remote Sensing with Machine Learning for Socio-Economic Analysis

Konstantinos Stavrakakis^a, David Pacios^b, Napoleon Papoutsakis^a, Nikolaos Schetakis^{ad}, Paolo Bonfini^a, Thomas Papakosmas^c, Betty Charalampopoulou^c, José Luis Vázquez-Poletti^b, Alessio Di Iorio^{ad}

^aALMA Sistemi Srl, Guidonia (Rome), 00012, Italy, ^bUniversidad Complutense de Madrid, 28040 Madrid, Spain, ^cGeosystems Hellas S.A., Athens, Greece, ^dQuantum Innovation Pc., 73100 Chania, Greece

ABSTRACT

EYE-Sense is a Web-GIS platform which allows for easy access to valuable socio-economic insights from Earth Observation (EO) data by offering a code-less approach. The platform enables users to access various EO parameters, such as atmospheric and water quality indexes, and night-light activity. Moreover, the platform's pre-trained computer vision models (Faster-RCNN, Mask-RCNN, and YOLO) empower users to detect objects such as, e.g., airplanes, ships, containers, and beach umbrellas, to address specific user-based tasks. To provide cost-efficiency, scalability, flexibility, and easy maintenance, EYE-sense adopts a serverless architecture, leading to up to 50.4% processing cost reduction when compared to traditional server-based solutions. By bridging the gap between data gathering and processing, EYE-Sense extends the reach of Earth observation data to a broader audience.

Keywords: Earth Observation, socio-economic activity, computer vision, web-gis, object detection, satellite images, serverless

1. INTRODUCTION

In a little more than a decade, the space sector has experienced considerable development throughout the world, with greater impacts on the larger economy, further boosted by both globalization and digitalization. In the context of the growing interest for the so-called “Space Economy”, there is an increasing awareness of the importance of the utilization of the available space assets for the global and local economy. There is a significant number of indirect parameters observable from space (EO parameters) that can be correlated to a variety of phenomena ranging from the impact of natural and man-made disasters on the macro/micro economy, to the progression of diseases such as Covid-19. Following on this example, one could investigate how classical environmental parameters (geographical, geomorphological, climatological and hydrogeological), and tracers of the human-induced impact on the environment (urbanization, pollution, heat) can be associated with economical parameters of human activities impacted by the epidemic, including transportation, industry, tourism and trades. Specific proxies tracing the evolution of the epidemic could be, e.g., the increase of heat production of crematories or the tourism indicators (such as beach occupancy). More generically, EO parameters may include monitoring of the levels of suspended pollutants, heat distribution around cities and industrial areas, percentage of transport container occupancy in the yard of industrial harbors, airports and the like. All these “observables” can be correlated to macro parameters, which in turn allow to study the progress of an epidemic and its impact on the economy at different scales. Although remote sensing methodologies have been – and are normally – employed to monitor local and global critical situations (from natural disasters¹ to environmental impact), their application to the monitoring of the economy has been limited to the agricultural³ sector. Moreover, the analysis method has been mostly limited to the visual inspection of satellite images². However, space technology combined with Machine Learning (ML) and Deep Learning (DL) could not only represent a powerful tool to monitor near real-time natural and man-made disasters, and economic trends but also to analyze complex data. Public Authorities, researchers, and private actors operating in the field of economy and finance are strongly interested in how space technologies can be linked to socio-economic factors and provide updated nowcasting and forecasting assessment. Up to now, there has been no systematic use of the space asset to monitor features from space that are linked with well-established econometric and epidemiology models. In fact, the correlation between what can be seen – and continuously monitored – from space and key econometric indicators of economic trends at different

geographical scales has not been addressed yet. EYE-Sense's great potential for scientific breakthroughs lies in its multidisciplinary approach, linking Geographic Economy with state-of-art engineering methodologies and tools. EYE-Sense can provide real time indicators of the spread of the epidemic, and its impact on the economy, in those countries and geographical areas experiencing Covid-19 outbreaks, where direct information and data are not fully reliable and/or provided with large delay.

2. MOTIVATION

A significant body of research has demonstrated the link between indicators of atmospheric^{4,5} and water quality^{6,7} and economic activity in an area. For example, cities with high level of economic activity are often subject to environmental regulations designed to reduce pollution and protect public health. Such regulations may impose costs on businesses, which in turn can affect economic activity. Areas with poor air quality may be less attractive to tourists and investors, which can have negative economic consequences as well. Regarding the water quality indicators, such as chlorophyll-a and suspended matter, clean and clear water is often an important factor in the attractiveness of a region in terms of tourism and recreational activities, such as swimming, boating, and fishing. High levels of chlorophyll-a or suspended matter can reduce water clarity, making it less attractive for tourists and reducing the economic benefits of these activities. Nighttime lights^{8,9} (NTL) captured by satellite images of the earth, have been used as an indicator of economic activity and they can be exploited to track the growth of urbanization and population density, as cities tend to have larger radiance than rural areas. The more urbanized an area is, the more likely it is to have higher levels of economic activity, such as trade, manufacturing, services, or tourism. EO parameters can be associated to Maritime traffic^{10,11} and aviation^{12,13} as a proxy for touristic activity, trade and exchange. Fluctuations in the number of containers^{14,15} located in a port or other depots can be associated with macro-economic trends. When the economic activity is high, the volume of goods being transported increases, resulting in more containers being moved through ports and other depots, and vice versa. In addition, container traffic can be used to monitor changes in consumer demand and production levels. We have steered on several potential candidates supported by the relevant literature. Namely, we considered: ship, aircraft, and container count (direct proxies of transport and trade activity), temporal evolution of beach umbrella count (proxy of touristic activity), water and atmosphere quality indicators (proxies for pollution), and NTL (proxy of human population or activities).

3. THE PLATFORM

EYE-Sense is a web-based decision support system for integrated socio-economic analysis based on geo-informatics, deep learning (computer vision), and information technology modeling. EYE-Sense was developed with the intention of assisting interdisciplinary research, and allowing an individual to access information that would traditionally require a strong technical background to obtain. As described in the earlier section, according to literature, there are several EO parameters that could provide insight regarding the economic trends of an area. Figure 1, presents a comprehensive summary of the EO parameters that are being provided as micro-services. In the following subsections, we will thoroughly describe the details of these microservices, the processing pipelines, the data sources, and the technical implementation details. Moreover, we demonstrate how our serverless computing architecture provides cost-efficient and scalable processing capabilities. Finally, we showcase the user-friendly interface that allows users to effortlessly access and analyze key indicators without prior technical expertise.

3.1 Data sources and processing chains

3.1.1 Ship Detection – Counting as Maritime Activity Indicator

This workflow is designed to provide the user with the number of ships detected in a specified Area Of Interest (AOI) and time interval. Our detection tool is based on YOLOv5¹⁶, a fast object detection model. Our model has been fine-tuned on an annotated dataset of Sentinel-1 images containing ship locations (SSDD)¹⁷. Sentinel-1 is a radar satellite mission developed by the European Space Agency (ESA) that provides high-resolution images of the Earth's surface. EYE-Sense can automatically detect ships in unseen Sentinel-1 images with a high degree of accuracy (97%). The detection pipeline initiates by acquiring a specific Sentinel-1 image. Next, the image is pre-processed via cropping, calibration, speckle

filtering, and terrain correction. Image calibration is necessary to adjust and standardize the satellite image to remove any distortions or anomalies caused by atmospheric conditions and sensor characteristics. Speckle filtering is used to remove noise from the images and enhance the signal-to-noise ratio. Terrain correction is necessary to account for any variations in the Earth's surface, which can affect the radar signal. The pre-processed image is then passed as input on the fine-tuned model. As an output to the user, EYE-Sense creates and displays time-series data depicting the object counts over a specified time interval.

Service Details	Ship Detection	Aircraft Detection	Container Detection	Beach Umbrella Detection	Water Quality Assessment	Atmosphere Quality Assessment	Night-Time Activity
Input	1. Area of Interest (geojson) 2. Dates of Interest	VHR Image	VHR Image	VHR Image	1. Area of Interest (geojson) 2. Dates of Interest	1. Area of Interest (geojson) 2. Dates of Interest	1. Area of Interest (geojson) 2. Dates of Interest
Output (Derived Indicators)	Time-Series of Ships detected in Area of Interest	1. Image with visualised detected objects 2. .txt file in coco format	1. Image with visualised detected objects 2. .txt file in coco format	1. Image with visualised detected objects 2. .txt file in coco format	Time-Series of water quality indicators (Chl-a, Turbidity etc.) for the given area of interest	Time-Series of atmospheric quality indicators (CO, HCHO etc) for the given area of interest	Time-Series of monthly night-light radiance for the given area of interest
Processing Infrastructure	Python environment using sentinelapi/GEE, snappy and pytorch	Python environment using OBBDetection and tensorflow	Python environment using pytorch	Python environment using mmdetectionv2	Python/Javascript Environment using GEE	Python/Javascript environment using GEE	Python/Javascript environment using GEE
Models Used	YOLOv5 for SAR Faster-RCNN for VHR images	Faster-RCNN	YOLOv5 / YOLOv7	Mask R-CNN	Band-based Statistical Models	Statistical Model	Statistical Model e.g. Prophet
Product	Aggregated number of Ships at given temporal resolution	Aggregated number of Aircrafts at given temporal resolution	Aggregated number of Containers and normalized container trend	Normalized trend of the number of umbrellas at a given temporal resolution	Monthly aggregated concentration s Chl-a & SPM	Monthly aggregated surface concentrations of key air pollutants (NO2, PM2.5 etc.)	Monthly average of night-lights upward mean radiance for an AOI
Satellite	Sentinel-1 / Sentinel-2 / VHR Images (Planet)	VHR Imagery (e.g. Planet)	VHR Imagery (WorldView3 or Planet) and Sentinel-2	VHR Imagery (WorldView-3 or drone shots)	Sentinel-2A & Sentinel-2B	Sentinel-5 (CAMS-ERA5)	Suomi-NPP
Instrument	SAR-C Band (Sentinel1) / MSI (Sentinel-2 and VHR)	MSI	MSI	MSI	MSI	Atmospheric numerical models and in-situ observations	VIIRS-DNB
Spatial Resolution	9m x 9m for Sentinel-1 10m x 10m for Sentinel-2 0.8-1.5 x 0.8-1.5 for VHR	(0.8-1.5)m x (0.8-1.5)m	(0.8-1.5)m x (0.8-1.5)m for VHR 10m x 10m for Sentinel-2	(0.8-1.5)m x (0.8-1.5)m	10m x 10m for Sentinel-2	11km x 11km	742m
Temporal	Every 3-5 days for Sentinel-1 / 1-10 days for Sentinel-2 / On demand VHR	Upon Request (even daily)	Upon Request (even daily) for VHR 1-10 Days for Sentinel-2	Upon Request (even daily)	1-10 days for Sentinel-2	Hourly	Monthly cloudfree composites
File Size	~800MB for Sentinel-1 ~600-700MB for Sentinel-2 ~100MB for VHR depending on the AOI	~100MB for VHR depending on the AOI	~600-700MB for Sentinel ~100MB for VHR depending on the AOI	~100MB for VHR depending on the AOI	~600-700MB for Sentinel-2	~100 KB	Monthly ~0.5GB
Database	Google Earth Engine / or 3rd-Party for VHR (e.g. UP42)	Creodias/ Google Earth Engine for Sentinel-2 or 3rd-party for VHR	Creodias/ Google Earth Engine for Sentinel-2 or 3rd-party for VHR	3rd Party (e.g. Planet - SKYSAT or WorldView-3)	Creodias / Google Earth Engine	Climate Data Store (cdaAPI) / Google Earth Engine	Google Earth Engine via Earth Observation Group

Figure 1. Details of the input/output for the provided services, along with technical implementation information.

3.1.2 Aircraft/ Trucks / Containers / Umbrellas Detection - Counting

This workflow is designed to detect objects in a Very High-resolution satellite image provided by the user. The user designates the type of object they intend to detect, selecting from four available options: ships, aircrafts, umbrellas, and containers. Depending on the type, the appropriately pre-trained model is selected to perform the detection task (Table 1). Different models are employed for each object type due to performance considerations and the ease of training specialized models, resulting in enhanced efficiency and effectiveness. For the detection of ships, and aircraft, the Faster R-CNN¹⁸ model pre-trained on the DOTA¹⁹ dataset is selected. Faster R-CNN is a state-of-the-art object detection algorithm composed of two elements: 1) a region proposal network which generates candidate object locations, and 2) a classification network which labels each candidate as “object” or “background”. The DOTA dataset contains annotated satellite images with various types of objects, including vehicles, ships, and aircraft. For the detection of umbrellas, a Mask R-CNN²⁰ model is used. Mask R-CNN is an extension to Faster R-CNN which adds a segmentation network to the architecture, allowing it to predict object masks in addition to bounding boxes and class labels. Finally, for the detection of containers, a custom YOLOv5 model is used. YOLOv5 relies on a single neural network to predict both bounding boxes and class probabilities for objects in an image. The custom YOLOv5 model is trained on a dataset of annotated satellite images including containers²². Once the appropriate model has been inferred, the image is parsed to detect the selected object. The results can be provided to the user either as an image with the detected objects highlighted, or as a .txt file in COCO format, a standard format for object detection results which includes bounding boxes, class labels, and confidence scores. Assuming that a user has multiple VHR images of the same region, the described workflow can be used iteratively to construct time-series data.

Table 1: Datasets and models used for each object’s detection.

Object	Dataset	Model	Accuracy
Aircraft	DOTA ¹⁹ , rareplanes ²¹	Faster-RCNN	99%
Ships	DOTA	Faster-RCNN	96%
Umbrellas	World-View-3 Annotated Images ²³	Mask-RCNN	75%
Containers	Dataset for detecting buildings containers cranes in satellite images ²²	YOLOv5	86%

3.1.3 Atmospheric quality analysis

The designed workflow aims to provide users with time series data pertaining to various atmospheric indicators, for a specified Area Of Interest (AOI). Users are required to input the AOI and select the desired atmospheric indicators from a list of six options: Sulfur dioxide (SO₂), Carbon monoxide (CO), Nitrogen dioxide (NO₂), Formaldehyde (HCHO), Aerosol Optical Depth at a wavelength of 550nm (AER_AI), and Ozone (O₃). Furthermore, users are given the option to choose the frequency at which the analysis should be conducted, such as on a weekly, biweekly, or monthly basis. Utilizing the Google Earth Engine²⁴ platform, Sentinel-5p satellite data is harnessed and processed to generate the requested time-series information. The Sentinel-5p satellite mission, developed by the European Space Agency (ESA), delivers high-resolution data on atmospheric composition. EYE-Sense extracts pertinent data for the specified atmospheric indicators from the Sentinel-5p satellite dataset. Employing the mean reducer offered by the Google Earth Engine (GEE), a single value (the mean value) is derived for the area of interest concerning each atmospheric indicator chosen by the user. Subsequently, these values are employed to produce a time-series for the atmospheric indicators at the specified frequency of analysis.

3.1.4 Water quality analysis

The Water Quality Analysis workflow provides users with EO parameters that may potentially correlate with the economic activity of a specified area. To initiate the analysis, users provide an AOI and time interval. The relevant multi-band spectral images produced by the Sentinel-2 mission (providing high-quality and frequent observations of the Earth's surface) are dynamically acquired. Specifically, we retrieve the data through the Google Earth Engine, which allows for efficient processing and analysis of extensive data volumes.

By adopting several literature prescriptions^{25,26} – based on the Sentinel-2 bands – we created indicators correlated with water quality parameters, including Chlorophyll-A, Cyanobacteria, Turbidity, Dissolved Organic Carbon, and Color Dissolved Organic Matter. The details of these formulas are presented in Table 2. The Normalized Difference Water Index (NDWI) is utilized to mask land and retain only water areas. The most suitable prescription is applied to compute the indicator value for each time-step, depending on the user's selection. Google Earth Engine's (GEE) mean reducer is employed to obtain a single value (the mean value) for the area of interest concerning each water quality indicator chosen by the user. These values are then harnessed to produce time-series data of the water quality indicators for the specified frequency of analysis. Users can visualize the derived time-series on an interactive map or download it in a .csv format.

Table 2: Water quality analysis information.

Parameter	Chlorophyl-A	Cyanobacteria	Turbidity	Color Dissolved Organic Matter	Dissolved Organic Carbon
Formula	$4.26 \left(\frac{B03}{B01} \right)^{3.94}$	$1.1553e5 \left(B03 \frac{B04}{B02} \right)^{2.38}$	$8.93 \left(\frac{B03}{B01} \right) - 6.39$	$537e^{-2.93 \frac{B03}{B04}}$	$432e^{-2.24 \frac{B03}{B04}}$
Unit	$\frac{mg}{m^3}$	$10^3 \frac{cell}{ml}$	NTU	$\frac{ml}{l}$	$\frac{mg}{l}$
Product	S2-L2A	S20-L2A	S2-L2A	S2-L1C	S2-L1C

3.1.5 Night-light activity analysis

The Nighttime lights (NTL) Analysis workflow providing users with an economic-related parameter indicator for a designated area, is arguably the most valuable EYE-Sense tool, and it is showcased in Section 4. To acquire high-quality NTL data, we rely on Visible Infrared Imaging Radiometer Suite (VIIRS) Day/Night Band (DNB)²⁷ satellite data. Specifically, we access the VIIRS DNB database through the Google Earth Engine platform, allowing us to effectively process and analyze large amounts of data. To initiate the analysis, users first set the Area of Interest (AOI) by uploading a file in GeoJSON format. VIIRS DNB data is subsequently retrieved from the Google Earth Engine data catalog, and preprocessed to remove any faulty values or interference, which is achieved by comparison against a supplementary dataset such as the VIIRS DNB straylight corrected dataset. Temporal aggregation is executed using Google Earth Engine's reducer functions like mean, max, and min. After completing data pre-processing and temporal aggregation, time-series analysis can be pursued, and the data for the chosen AOI can be either visualized or exported in a .csv format.

3.2 Platform architecture

In this section, we provide a technical implementation overview of each microservice. EYE-Sense consists of six microservices, each designed to handle a specific type of analysis. We describe the backend frameworks, external libraries, and tools used in each microservice, as well as any design considerations and challenges faced during the development process. An overview of these microservices, along with a small description of their workflows, is depicted in Figure 2. The microservices have been extensively presented in Section 3.1 – In summary: Ship Detection, employs Sentinel-1 images to identify the presence of ships within a specified area; Very High-Resolution Object Detection, is capable of detecting aircrafts, containers, umbrellas, and ships, within high-resolution satellite imagery; The remaining three microservices, i.e., NTL Analysis, Atmospheric Analysis, and Water Quality Analysis, all leverage Google Earth Engine. NTL Analysis infers urbanization and economic activity, Atmospheric Analysis infers air quality, and Water Quality Analysis infers water pollution assessment. The final microservice encompasses the frontend, offering an intuitive, user-friendly interface that enables users to engage with the other microservices and visualize results in real-time. Through the

incorporation of modern visualization libraries and a responsive design, our platform aims to deliver a robust tool for comprehensive data analysis and exploration.

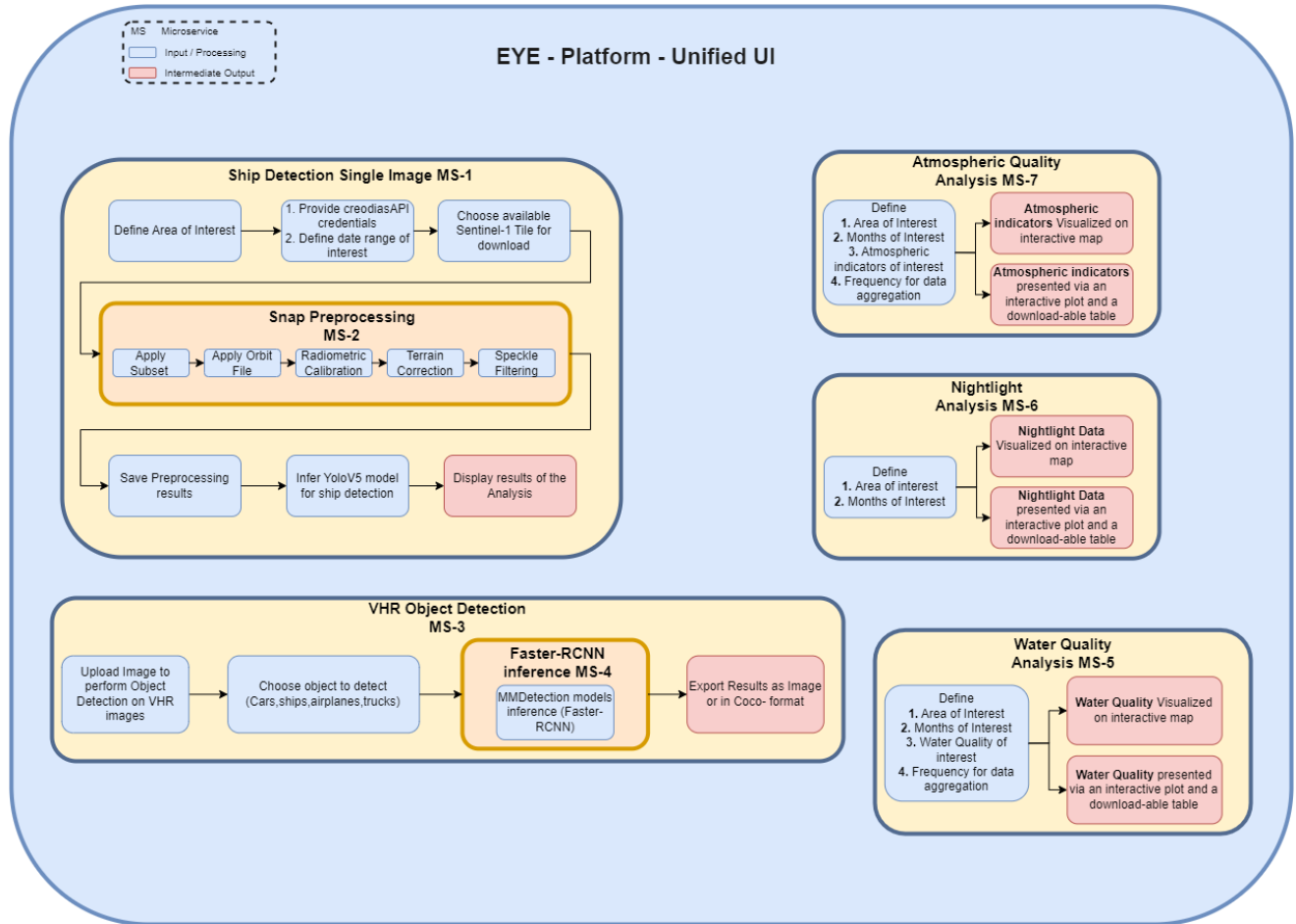


Figure 2. Platform architecture.

3.2.1 Microservices Technology Stack

Let's now delve into the technology stack employed for the development of each individual microservice. A schematic depiction can be seen in Figure 3. The Ship Detection microservice serves as a REST API that offers ship detection functionality for a specified area. It utilizes the fastAPI backend framework²⁸, a modern Python web framework is renowned for its exceptional performance and user-friendliness. The API calls for the Sentinelsat-API²⁹ to obtain Synthetic Aperture Radar (SAR) images from Sentinel-1, which are subsequently preprocessed via the snappy toolkit³⁰ to furnish the required inputs for a YOLOv5 object detection model implemented in PyTorch. The Very High Resolution (VHR) Object Detection microservice, -functions as a REST API for detection in VHR images. The backend framework employs fastAPI. For umbrella detection in VHR images, we utilized the Mask-RCNN model implemented in PyTorch. A YOLOv5 model, also implemented in PyTorch, was employed for container detection. In the case of aircraft and ship detection, we used a Faster-RCNN model from the MMDetection framework³¹, a widely-used open-source object detection toolbox. NTL Analysis, Atmospheric Analysis, and Water Quality Analysis all exploit Google Earth Engine to acquire data from distinct datasets. Despite their varying data sources, these microservices are structured similarly and constructed using

fastAPI. The output results from each microservice are presented as a pandas dataframe in JSON format, offering a flexible and user-friendly format for data analysis and visualization.

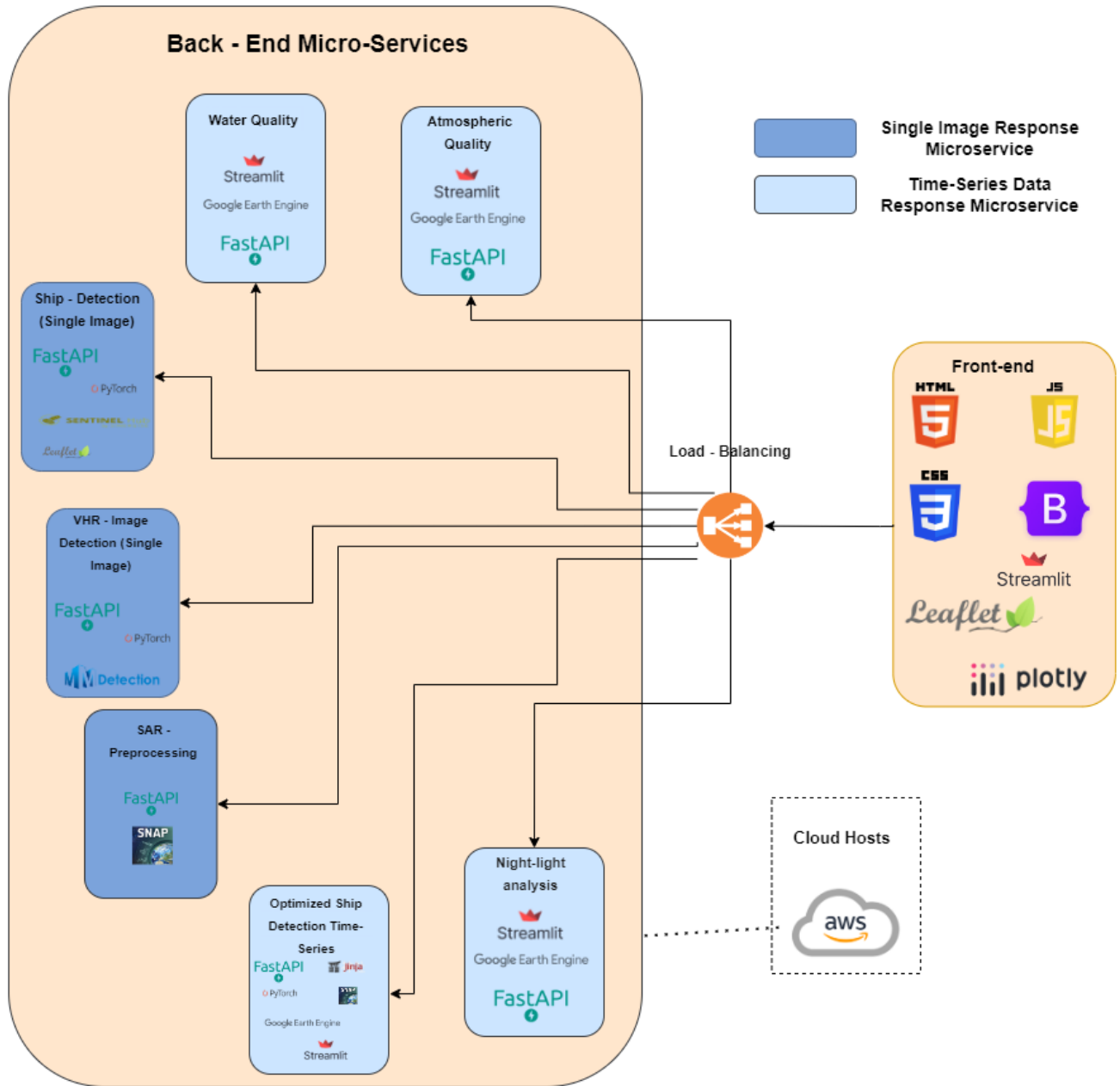


Figure 3. Technology Stack Details.

One of the main design considerations for all these five microservices was scalability. The ability to process large volumes of data quickly and accurately was a top priority, as the datasets used in these microservices can be quite large. We also focused on making the microservices modular and easy to integrate with other applications and platforms. One challenge faced during the development of these microservices was optimizing the data processing pipeline to ensure speed.

Additionally, working with large datasets can be computationally intensive, so we had to carefully balance computational resources with the need for accurate results. Moreover, ensuring that the models in the Ship Detection and VHR Object Detection microservices remained accurate while maintaining real-time performance was another challenge that we overcame through model selection, fine-tuning, and optimization. The Frontend is built on top of Streamlit³², a Python library that allows for the creation of interactive web applications. Several visualization libraries were utilized, such as Plotly³³ and LeafletJS³⁴, to create an intuitive and user-friendly UI experience. Plotly allows for the creation of interactive charts and graphs, while LeafletJS provides a flexible mapping library for visualizing geospatial data. The frontend provides a streamlined and intuitive way for users to interact with the data and makes it easy to explore different visualizations and insights. One of the main design considerations for the Frontend was creating a responsive and visually appealing interface that would be easy for users to navigate. We focused on creating a design that would be both functional and aesthetically pleasing, while also ensuring that the interface was intuitive and easy to use. We also had to ensure that the interface was responsive and performant, even when dealing with large datasets or complex visualizations.

3.2.2 Leveraging Serverless Computing

In this section, we discuss the serverless architecture used to efficiently run the computer vision models that are crucial to our platform. Leveraging serverless computing allows us to seamlessly scale the processing of these models, enabling faster and more accurate results for our users. Below, we briefly describe the fundamentals of Serverless Computing and then we delve into the specific benefits and implementation details of our serverless approach. Serverless computing, also known as Function as a Service (FaaS), is a cloud computing model that allows developers to focus solely on coding and deploying their applications without the need to manage underlying servers or infrastructure^{35,36}. The cloud provider handles server management, capacity planning, and scaling, making it more efficient and cost-effective for developers^{37,38,39,40,41}. Popular commercial FaaS platforms include AWS Lambda, Google Cloud Functions, and Microsoft Azure Functions, while open-source alternatives include Apache OpenWhisk, OpenLambda, and Knative. Serverless computing is well-suited for deploying and executing scientific workloads, as it allows for optimal resource provisioning and function chaining^{42,43}. We opted for AWS Lambda, as it enables code execution without managing infrastructure components while defining events that activate the functions⁴⁴. AWS Lambda executes each function within a dedicated container, which is then executed on a multi-tenant cluster of machines managed by AWS. The total runtime of a function in AWS Lambda includes the execution of the code itself and the initialization performed by Lambda⁴⁵. The code is executed within a container deployed for this purpose, which will be reused if the function is triggered again within the next 15 minutes, reducing the initialization time⁴⁶. EYE-Sense utilizes AWS Lambda and serverless architecture to develop an object detection system capable of inferring the trained Computer-Vision models. For instance, our YOLOv5 models are implemented through Lambda functions, enabling parallel execution and optimal resource utilization. To handle large files exceeding 1GB in size, we employed AWS Elastic File System (EFS) and integrated it with the API Gateway. The API Gateway serves as an intermediary for securely accessing the EFS files, ensuring optimal performance and scalability while maintaining a high level of security. Our implementation also considers the "cold start" time, which occurs when a new instance of the Lambda function is initiated, and the EFS file system needs to be mounted. We carefully analyzed the cold start time to ensure efficient execution of the Lambda function. Our object detection system utilizes an Amazon S3 bucket as the input for processing images, with the API Gateway managing requests and triggering the YOLOv5 Lambda function. The YOLOv5 Lambda function then processes images, performs object detection, and deposits the resulting output in a designated S3 bucket, ensuring seamless and efficient operations of our Platform. Using the EFS-Lambda model provides several advantages over local computing, including efficient parallelism, scalability, automatic scaling, and a pay-as-you-go pricing model, which significantly reduces operational costs and simplifies infrastructure management. One approach we employ to minimize the costs associated with object detection tasks is to avoid the use of Amazon S3 or Elastic File System (EFS) storage when possible. Specifically, in cases where the available 512 MB of temporary storage provided by Lambda instances is sufficient for the task at hand, we opt not to use S3 or EFS storage. The motivation behind this approach is to reduce costs, as we will argue in the remainder.

We compared the actual cost of using AWS Lambda functions with or without S3 and EFS storage versus using Amazon EC2 with S3 storage. Amazon EC2 are virtual computing environments that allow users to rent virtual servers, also known as instances, to run their applications. We tested these three deployment configurations for a simple scenario. We inferred a YOLOv5 model on 1000 images with an average size of 50 MB each. For the AWS Lambda functions scenarios we allocated 300 MB of memory and we had a duration of 40 seconds per image inference and for the EC2 scenario we have a t3.medium instance with 2 vCPUs and 4 GB RAM. We considered the following factors for cost estimation:

1. Storage: We stored the 1,000 images for processing. Storing Prices were calculated according to Amazon S3 Pricing.⁴⁷ (\$0.023/GB)

2. Processing: We estimated the cost of processing the images using AWS Lambda functions and an EC2 instance with the same processing time. Prices were calculated according to Amazon EC2 On-Demand Pricing⁴⁸ and AWS Lambda Pricing⁴⁹, which are \$0.0416/hour for EC2 t3.medium and \$0.20 per 1M requests, \$0.0000166667 for every GB-second for AWS lambdas.

Table 3. Actual cost estimates for each scenario (data transfer were considered negligible for all the scenarios).

Scenario	Storage Cost	Processing Cost	Total Cost
Lambda with S3 and EFS Storage	\$1.15 (50GB)	\$0.1956	\$1.3456
EC2 Instance with S3 Storage	\$1.15 (50GB)	\$0.394 (9.44 hours)	\$1.544
Lambda without S3 Storage	Negligible	\$0.1956	\$0.1956

Based on the data presented in Table 3, it is evident that processing 1,000 images using AWS Lambda functions without S3 storage incurs a cost of approximately \$0.20, which is significantly lower than the cost of using Amazon S3 for storage and Amazon EC2 instances, which is approximately \$1.54. The implementation of AWS Lambda functions can provide cost savings from 12.5% when we use S3 storage up to 85% percent when we do not use S3 storage in our Lambdas. At the same time, they offer the benefits of parallel processing, resulting in higher scalability and throughput when compared to the use of Amazon EC2 instances. While our testing and implementation were focused on the YOLOv5 model, similar performance boost and cost-effectiveness is to be expected when inferring our other models using AWS Lambda with or without temporary storage and parallel processing. By leveraging the benefits of AWS Lambda, EYE-Sense can efficiently parallelize the processing of image datasets, resulting in improved scalability and faster processing times. Additionally, by using temporary storage within the Lambda function, we eliminate the need for additional storage costs, further reducing the overall cost of image inference.

3.3 User Interface

EYE-Sense's user interface (UI) is designed to offer an intuitive and seamless experience for users. In this section, we discuss the layout and features of our UI, which can be accessed through a web browser. Starting with an overview of the Landing Page (Figure 4), it presents a comprehensive summary of the platform's capabilities. Upon logging in, users encounter a navigation menu that facilitates easy access to each of our workflows, allowing for smooth transitions between services. Besides the navigation menu, the landing page showcases an example of the platform's usage through interactive plots generated from data provided by the platform. The plots illustrate a scenario executed for the city of Heraklion (Greece) area, visualizing results for ship detection analysis, atmospheric quality analysis, water quality analysis, and nightlight activity analysis. These interactive plots enable users to explore and evaluate data in a user-friendly and intuitive manner, demonstrating the potential of our platform. Overall, the landing page serves as a direct introduction to our platform, highlighting the scope and versatility of its capabilities and allowing users to effortlessly engage in the exploration.

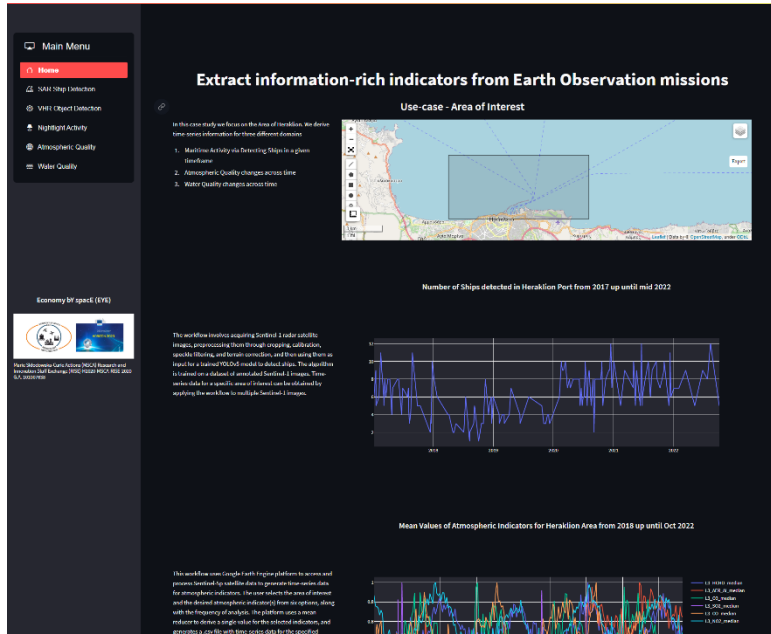


Figure 4. Landing Page – UI.

3.3.1 Ship detection workflow – UI

Our platform offers a highly effective workflow for identifying ships in synthetic aperture radar (SAR) images using a YOLOv5 object detection model. Users provide their SentinelAPI credentials and choose a time-frame of interest, upon which our system searches for available Sentinel-1 tiles within the specified timeframe (as illustrated in Figure 5, *left*). The user can then select their preferred preprocessing steps and initiate the ship detection algorithm (Figure 5, *right*). Upon completion of the detection algorithm, users receive an email containing a .zip file with the processed images, each displaying bounding boxes that indicate detected ships, as well as multiple *coco.txt* files containing the results in COCO format. Figure 6 displays the [zoomed-in] result of ship detection on an image. Our user-friendly interface streamlines the entire process, enabling users to effortlessly search for SAR images, customize detection parameters, and obtain comprehensive output files with minimal effort.

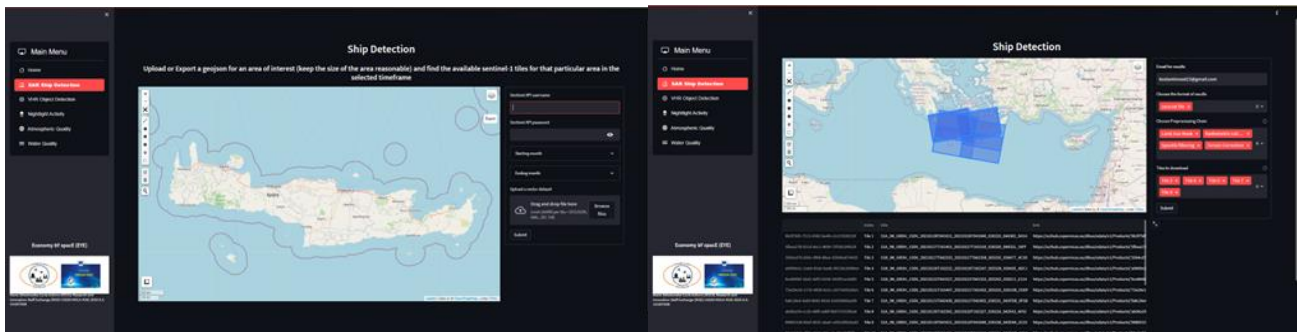


Figure 5. Ship detection workflow UI – area selection.

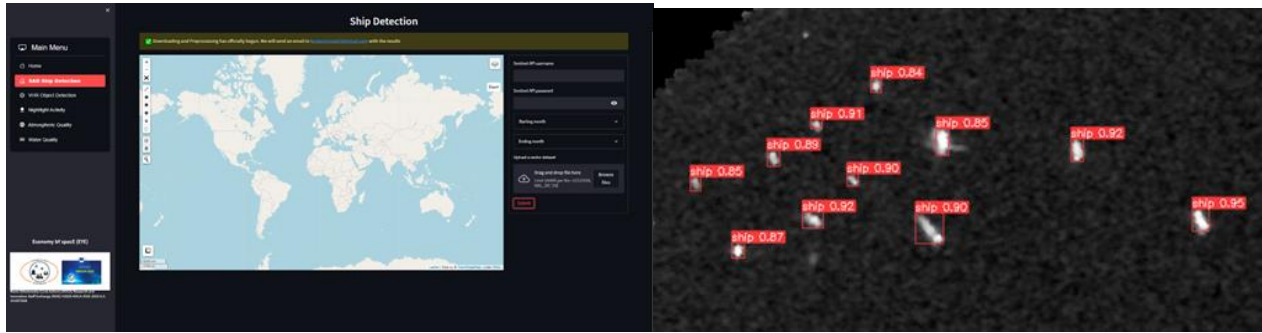


Figure 6. Ship detection workflow UI – detection results.

3.3.2 VHR object detection - UI

A potent workflow is available for detecting a broad spectrum of objects within satellite images with very high resolution (VHR). Images can be uploaded by the user and the targets of interest, namely aircrafts, ships, containers, or umbrellas, can be selected, as shown in Figure 7. Once the object detection algorithm has completed, the user is provided with the results both as an image with bounding boxes indicating the detected object(s) and as a .txt file in COCO format, allowing for easy integration with other software or analysis tools. An example of the output images of our object detection models can be seen in Figure 8. The interface simplifies customization of detection parameters, uploading and processing of large data volumes, and generation of detailed output files with minimal effort.

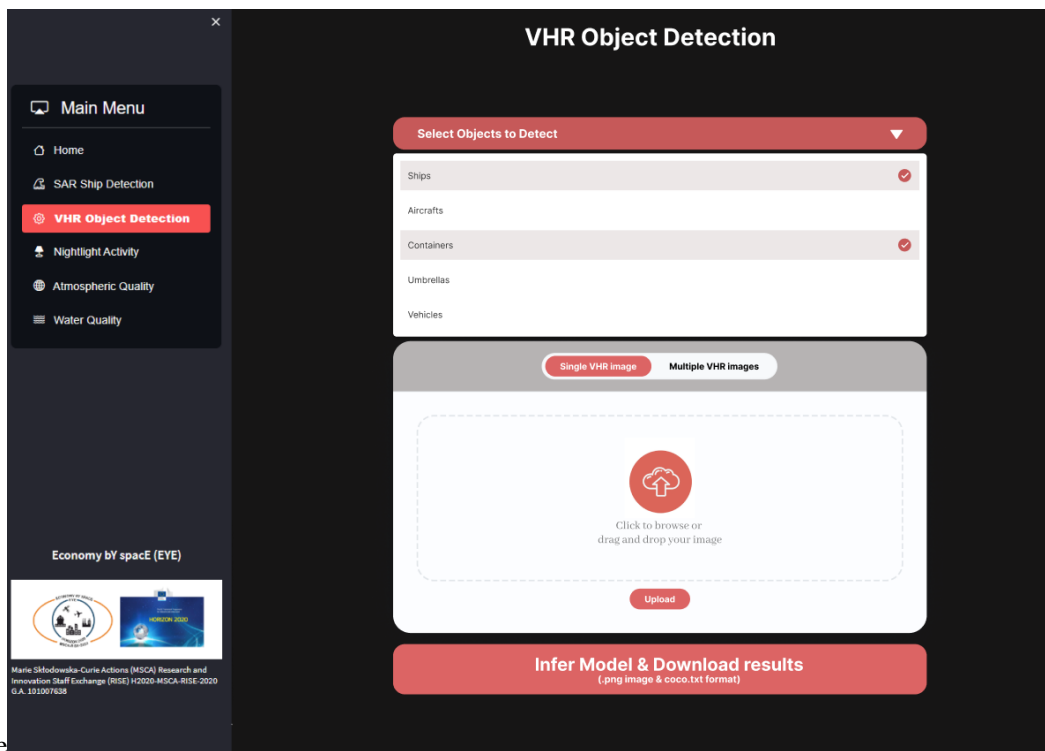


Figure 7. VHR object detection workflow UI.

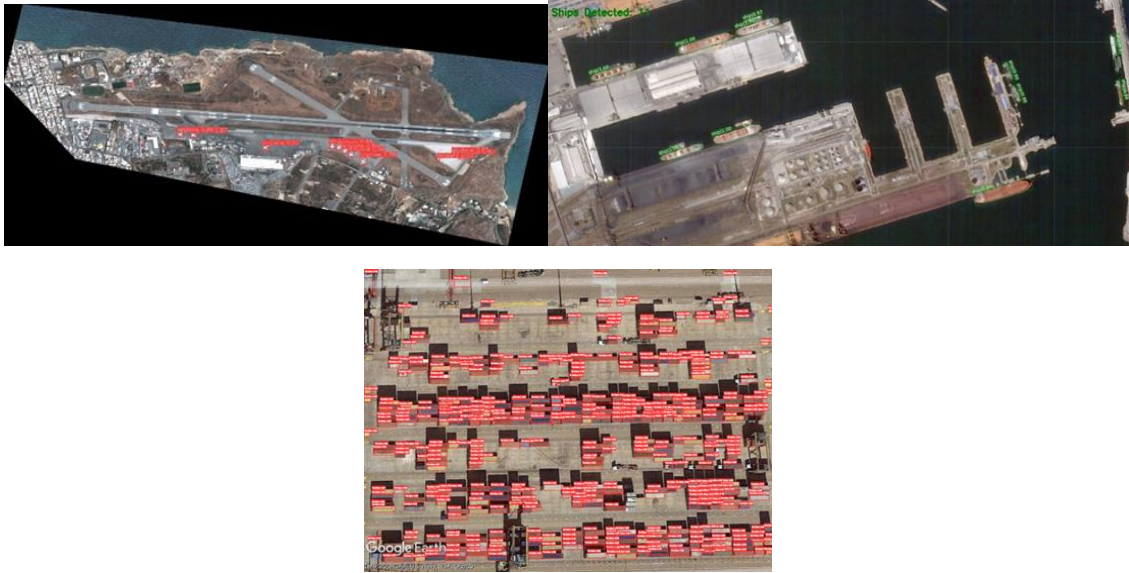


Figure 8. Object detection annotated results for aircrafts, ships and containers.

3.3.3 Atmospheric quality analysis – UI

EYE-Sense provides an efficient and user-friendly workflow for accessing and analyzing atmospheric indicators from Sentinel-5P data using Google Earth Engine. The user simply specifies the area of interest, the time interval for analysis, and the desired atmospheric indicators (Figure 9, *top left*). After retrieving and processing the pertinent data from the Earth Engine database, EYE-Sense offers users the choice to view the visualized results on an interactive map (Figure 9, *top right*) or extract raw data in .csv format. For those who prefer visualizing the data, we also offer an interactive plot graph in our UI (Figure 9, *bottom*). The user-friendly interface provided by our platform simplifies the process of exploring and analyzing atmospheric indicator data, allowing users to customize analysis parameters and generate detailed output files with ease.

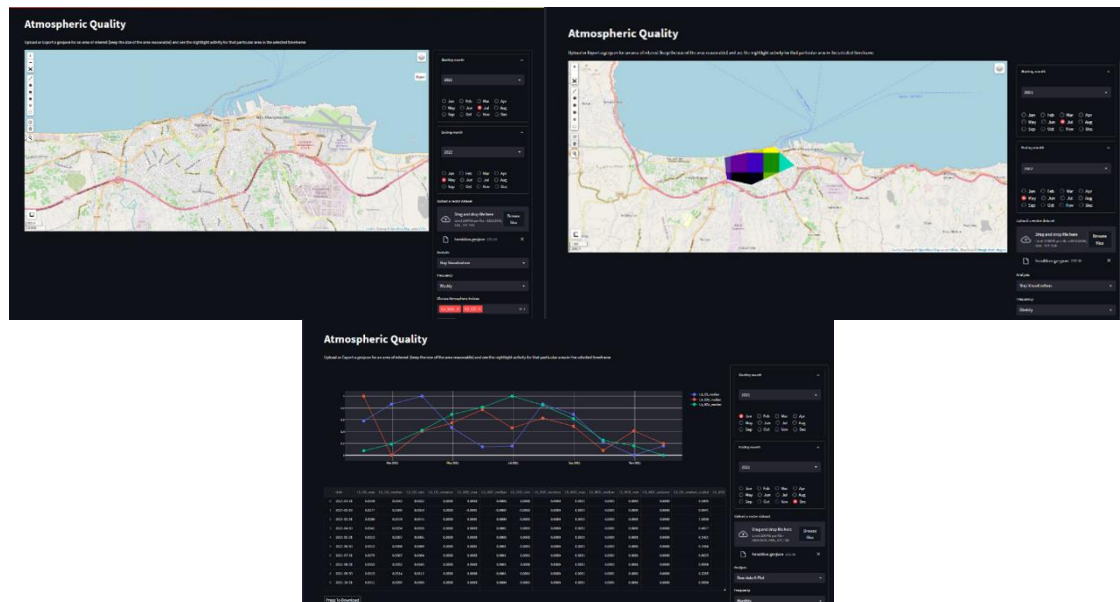


Figure 9. Atmospheric quality analysis - UI.

4. USE CASE

In this section, we will focus on a specific use-case scenario as a representative example of the platform’s functionalities and workflow. In this scenario, a policy-maker responsible for overseeing tourism in a particular area lacks the technical know-how and resources necessary to extract valuable insights from the massive amount of satellite data. EYE-Sense can assist the user by converting satellite data into more straightforward insights. The overall workflow of the platform (Figure 10) involves three steps. Initially, the user chooses a service (EO parameter) from the list presented in Figure 1. Afterwards, they choose a particular area of interest and lastly, they define a time frame that pertains to the scenario they want to explore.

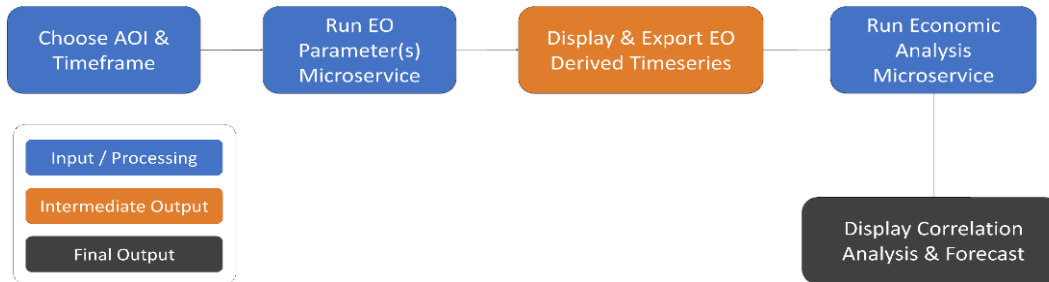


Figure 10. EYE web-platform workflow.

To illustrate, EYE-Sense will be utilized in the municipality of Heraklion, Greece, known for its thriving tourist industry. The aim is to offer insights into touristic activities using three EO parameters: nighttime lights (NTL), atmospheric and oceanic quality. Our objective is to establish a correlation between ground aviation-transportation data obtained from the airport of Heraklion and the previously mentioned EO parameters. In Figure 11 (top), the normalized aviation data is presented in the form of time series, showcasing the monthly arrival statistics of international passengers (green line) arriving to Heraklion's airport from 01/2017 to 12/2021. Our next step was to define an Area Of Interest (AOI) for which we will obtain EO parameters for the same time period as the aviation data. We opted to concentrate on the Heraklion coastal zone as the AOI, depicted in Figure 11 (bottom), as it is a prominent touristic area. For that area, we obtained the normalized NTL radiance time series and plotted it against the aviation data, as shown in Figure 11 (top).

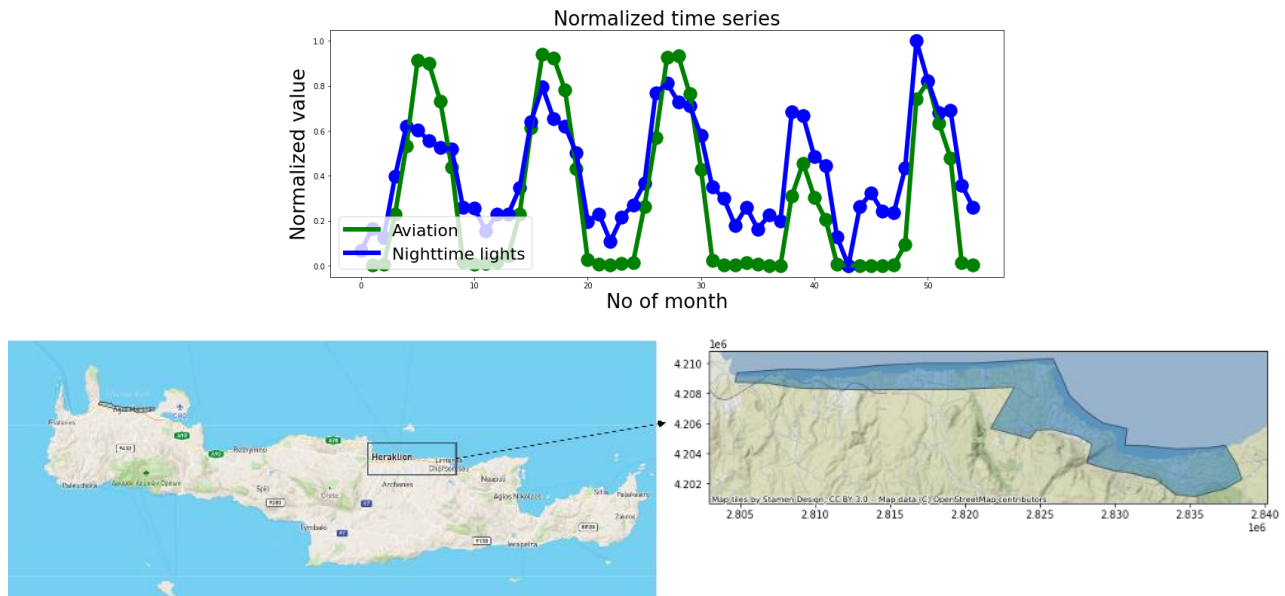


Figure 11. Top: normalized time series of Aviation data (green line) and NTL radiance (blue line) obtained from 2017 to 2022. Bottom: Heraklion municipality. The grey polygon highlights the Area of interest.

In order to assess the degree of association between the two time series we use statistical techniques such as the Pearson, Kendall and Spearman correlation coefficients, which indicate the strength and direction of the linear relationship between the two time series. The correlation coefficients between Aviation data and A. Nighttime-lights B. Atmospheric and V. Oceanic data are presented in Table 5. The first three rows present the Pearson, Kendall, and Spearman correlation coefficients for the normalized time-series. The next three rows present only the Pearson coefficient – however, this time, to alleviate the noise and smooth the time-series, we first transformed each time series in the following 3 ways:

1. Applied a rolling average with a window of 3 (months).
2. Decomposed each time series into its trend components.
3. Used the differencing method to transform them into stationary time series.

Subsequently, we estimated the correlation coefficient between each resulting time series and present our findings in Table 4. The correlation values of 0.92 (Pearson) , 0.71 (Kendall) and 0.87 (Spearman) indicate a high degree of correlation between Aviation and night time radiance.

Table 4. Correlation metrics between Aviation data and 1. NTL 2. Maritime 3. Atmospheric and 4. Oceanic data.

Correlation coefficient	Time series transformation method	Time series							
		Nighttime lights	Atmospheric data				Oceanic data		
			HCHO	O3	CO	NO2	CDEOM_p50	Turbidity p50	Chl p50
Pearson	Normalized	0.92	0.77	-0.47	-0.28	0.72	0.5	0.57	0.47
Kendall	Normalized	0.71	0.56	-0.29	-0.32	0.45	0.34	0.37	0.25
Spearman	Normalized	0.87	0.73	-0.48	-0.45	0.63	0.47	0.51	0.37
Pearson	Rolling averages	0.95	0.86	-0.51	-0.39	0.73	0.64	0.64	0.51
Pearson	Trend component	0.95	0.87	-0.47	-0.28	0.74	0.6	0.66	0.54
Pearson	Stationary	0.82	0.89	-0.59	-0.46	0.74	0.62	0.68	0.52

By applying linear regression analysis, we can examine the relationship between the dependent variable (aviation) and the rest of independent variables (NTL, atmospheric and oceanic data). Using Ordinary Least-Squares (OLS) regression we obtain the results listed in Table 5, illustrating the importance of each feature in predicting Aviation data. In this case, the NTL had a significant positive coefficient, indicating it can be used as a predictor for aviation data. Other factors, such as atmospheric pollutants (HCHO, O3, CO, NO2) and oceanic pollution (CDOM_p50, Turbidity p50, Chl p50), had limited or marginal significance, meaning the impact of touristic activity on them.

Table 5: Feature importance for the prediction of aviation data.

	Coef	Std err	t	P> t	[0.025	0.975]
Const	-0.136	0.088	-1.561	0.136	0.321	0.047
NTL	0.610	0.179	3.414	0.003	0.235	0.986
HCHO_median	0.131	0.147	0.891	0.385	-0.178	0.440
O3_median	0.137	0.124	1.108	0.282	-0.123	0.398
CO_median	0.007	0.137	0.051	0.960	-0.281	0.296
NO2_median	0.227	0.171	1.332	0.199	-0.132	0.587
CDOM_p50	-1.011	0.555	-1.823	0.085	-2.177	0.154
Turbidity_p50	1.425	0.756	1.884	0.076	-0.164	3.014

Chl_p50	-0.720	0.376	-1.918	0.071	-1.510	0.069
---------	--------	-------	--------	-------	--------	-------

In simpler terms, this analysis helps a user understand how touristic activity is associated with NTL and can possibly influence atmospheric and water quality in the Heraklion municipality. The analysis shows that nighttime light radiance, can be an indicator of tourism activity while other EO parameters like atmospheric and oceanic pollution play a less clear role. This information can help local authorities and businesses make informed decisions about tourism development and environmental policies without having to wait for ground data. This section showcases the procedures and resultant outputs that can be generated via EYE-Sense when the user provides the following inputs 1. Ground data 2. AOI and 3. Time period.

5. CONCLUSION

EYE-Sense web-platform presents a novel, important, and user-friendly solution for acquiring and analyzing satellite data from various sources without requiring any coding expertise. Its versatility is demonstrated in its diverse range of supported workflows, which can be used in multiple domains such as ship and object detection, nightlight activity, atmospheric data, and water quality data. Platform's implementation, based on a microservices architecture and a responsive UI developed, highlights its adaptability, scalability, and ease of use. The platform's innovative approach to managing intricate tasks with high performance is further emphasized by utilizing serverless computing (for the computer vision component). The practical significance and impact of the EYE-Sense platform are demonstrated by its application to the Heraklion municipality use-case. By correlating satellite data with economic activity and environmental factors, the platform enables users to gain valuable insights into regional tourism and its effects on the environment. Such information is crucial for local authorities, businesses, and stakeholders in making informed decisions regarding tourism development and environmental policies. EYE-Sense's code-less approach to satellite data acquisition and analysis lowers the barrier of entry for users with diverse backgrounds and expertise, opening up opportunities for a wider audience to leverage the power of satellite data. The platform's novelty, importance, and ease of use make it a significant contribution to the field and an asset for researchers, policymakers, and businesses alike.

6. ACKNOWLEDGEMENTS

K.S. , D.P. , N.P., N.S. , T.P. , B.C. , J.L.V.-P, and A.D.I. would like to acknowledge funding support from the European Union's Horizon 2020 research and innovation programme EYE (<https://cordis.europa.eu/project/id/101007638>) under the Marie Skłodowska-Curie grant agreement No.101007638

REFERENCES

- [1] Schneider, D. J., Randall, M., and Parker, T., "Volcview: A Web-Based Platform for Satellite Monitoring of Volcanic Activity and Eruption Response," in Proceedings of the Fall Meeting 2014, Abstract ID IN41D-05, Presented at the American Geophysical Union Conference, San Francisco, CA, USA, (2014).
- [2] Morteza Khazaei, Saeid Hamzeh, Najmeh Neysani Samani, Arnab Muhuri, Kalifa Goïta, Qihao Weng, "A web-based system for satellite-based high-resolution global soil moisture maps," Computers & Geosciences,170, 105250 (2023). DOI: <https://doi.org/10.1016/j.cageo.2022.105250>.
- [3] Kourgialas, N. N., Hliaoutakis, A., Argyriou, A. V., Morianou, G., Voulgarakis, A. E., Kokinou, E., Daliakopoulos, I. N., Kalderis, D., Tzerakis, K., Psarras, G., Papadopoulos, N., Manios, T., Vafidis, A., and Soupios, P., "A web-based GIS platform supporting innovative irrigation management techniques at farm-scale for the Mediterranean island of Crete," Science of The Total Environment, 842,156918 (2022).DOI: <https://doi.org/10.1016/j.scitotenv.2022.156918>.

- [4] Vibhooti Shukla and Kirit Parikh, "The environmental consequences of urban growth: cross-national perspectives on economic development, air pollution, and city size", *Urban Geography*, 13(5), 422-449 (1992). DOI: 10.2747/0272-3638.13.5.422
- [5] Shaw, D., Pang, A., Lin, C. C., & Hung, M. F., "Economic growth and air quality in China. *Environmental economics and policy studies*", 12(3), 79-96(2010).
- [6] Muyibi, S.A., Ambali, A.R. & Eissa, G.S., "The Impact of Economic Development on Water Pollution: Trends and Policy Actions in Malaysia.", *Water Resour Manage* 22, 485–508 (2008). DOI: <https://doi.org/10.1007/s11269-007-9174-z>
- [7] An Hao Cai, Yadong Mei, Junhong Chen, Zhenhui Wu, Lan Lan, Di Zhu, "An analysis of the relation between water pollution and economic growth in China by considering the contemporaneous correlation of water pollutants," *Journal of Cleaner Production*, 276, 122783(2020). DOI: <https://doi.org/10.1016/j.jclepro.2020.122783>.
- [8] Keola, S., Andersson, M., and Hall, O., "Monitoring Economic Development from Space: Using Nighttime Light and Land Cover Data to Measure Economic Growth," *World Development* 66, 322-334, 0305-750X(2015). DOI: <https://doi.org/10.1016/j.worlddev.2014.08.017>
- [9] Kulkarni, R., Haynes, K. E., Stough, R. R., and Riggle, J. D., "Revisiting Night Lights as Proxy for Economic Growth: A Multi-Year Light Based Growth Indicator (LBGI) for China, India and the U.S.," *GMU School of Public Policy*, 2011-12(2011). DOI: <http://dx.doi.org/10.2139/ssrn.1777546> .
- [10] Akbulaev, N. and Bayramli, G., "Maritime transport and economic growth: Interconnection and influence (an example of the countries in the Caspian sea coast; Russia, Azerbaijan, Turkmenistan, Kazakhstan and Iran)," *Marine Policy* 118, 104005, 0308-597X(2020)., DOI: <https://doi.org/10.1016/j.marpol.2020.104005> .
- [11] Fratila (Adam), A., Gavril (Moldovan), I. A., Nita, S. C., and Hrebenciuc, A., "The Importance of Maritime Transport for Economic Growth in the European Union: A Panel Data Analysis," *Sustainability* 13(14), 7961(2021). DOI: <https://doi.org/10.3390/su13147961>
- [12] Zhang, F. and Graham, D. J., "Air transport and economic growth: a review of the impact mechanism and causal relationships," *Transport Reviews* 40(4), 506-528(2020). DOI: <https://doi.org/10.1080/01441647.2020.1738587>
- [13] Ishutkina, M. and Hansman, R. J., "Analysis of Interaction between Air Transportation and Economic Activity," in *The 26th Congress of ICAS and 8th AIAA ATIO*,(2008). DOI: 10.2514/6.2008-8888
- [14] Guoqiang, Z., Ning, Z., Zhenqiang, W., and Qingyun, G., "Container ports development and regional economic growth: An empirical research on the Pearl River Delta region of China," (2005).
- [15] Özer, M., Canbay, Ş., and Kırca, M., "The impact of container transport on economic growth in Turkey: An ARDL bounds testing approach," *Research in Transportation Economics* 88, 101002, 0739-8859(2021). DOI: <https://doi.org/10.1016/j.retrec.2020.101002>
(<https://www.sciencedirect.com/science/article/pii/S0739885920302006>)
- [16] Jocher G, Chaurasia A, Stoken A, Borovec J, NanoCode012, Kwon Y, Michael K, 22 November 2022. "ultralytics/yolov5: v7.0 - YOLOv5 SOTA Realtime Instance Segmentation." <https://zenodo.org/record/7347926#.ZCXIYnZBwQ8>
- [17] Zhang, T., Zhang, X., Li, J., Xu, X., Wang, B., Zhan, X., Xu, Y., Ke, X., Zeng, T., Su, H., Ahmad, I., Pan, D., Liu, C., Zhou, Y., Shi, J., & Wei, S., "SAR Ship Detection Dataset (SSDD): Official Release and Comprehensive Data Analysis." *Remote Sensing*, 13(18), 3690 (2021). DOI: <https://doi.org/10.3390/rs13183690>
- [18] Ren, S., He, K., Girshick, R., & Sun, J., "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks." *Advances in Neural Information Processing Systems*, 28, 91-99 (2015).
- [19] Xia, G. S., Bai, X., Ding, J., Zhu, Z., Belongie, S., Luo, J., ... & Zhang, L. "DOTA: A Large-Scale Dataset for Object Detection in Aerial Images." *IEEE Conference on Computer Vision and Pattern Recognition*, 3974-3983(2018). DOI: <https://doi.org/10.48550/arXiv.1711.10398>
- [20] He, K., Gkioxari, G., Dollár, P., & Girshick, R., "Mask R-CNN." *The IEEE International Conference on Computer Vision (ICCV)*, 2980-2988 (2017). DOI: <https://doi.org/10.48550/arXiv.1703.06870>
- [21] Shermeyer, J., Hossler, T., Van Etten, A., Hogan, D., Lewis, R., & Kim, D., "Rareplanes: Synthetic Data Takes Flight." In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV 2021)*, 207-217 (2021). DOI: <https://doi.org/10.48550/arXiv.2006.02963>
- [22] Miyazaki, H., "A Dataset for Detecting Buildings, Containers, and Cranes in Satellite Images." *IEEE Dataport*, 1-4, (2022). DOI: <https://dx.doi.org/10.21227/7yfp-9p87>

- [23] Osmar Luiz Ferreira de Carvalho, Osmar Abílio de Carvalho Júnior, Anesmar Olineo de Albuquerque, Nickolas Castro Santana, DÍbio Leandro Borges, Argelica Saiaka Luiz, Roberto Arnaldo Trancoso Gomes, and Renato Fontes Guimarães, "Multispectral panoptic segmentation: Exploring the beach setting with worldview-3 imagery.", *International Journal of Applied Earth Observation and Geoinformation*, 112, 102910,1569-8432(2022) . DOI: <https://doi.org/10.1016/j.jag.2022.102910>.
- [24] Gorelick, N., Hancher, M., Dixon, M., Ilyushchenko, S., Thau, D., & Moore, R., "Google Earth Engine: Planetary-scale geospatial analysis for everyone.", *Remote Sensing of Environment*, 202, 18-27(2017). DOI: <https://doi.org/10.1016/j.rse.2017.06.031>.
- [25] Toming, Kaire, Tiit Kutser, Alo Laas, Margot Sepp, Birgot Paavel, and Tiina Nõges., "First Experiences in Mapping Lake Water Quality Parameters with Sentinel-2 MSI Imagery", *Remote Sensing* 8(8),640 (2016). DOI: <https://doi.org/10.3390/rs8080640>
- [26] Potes, M., Rodrigues, G., Penha, A. M., Novais, M. H., Costa, M. J., Salgado, R., and Morais, M. M., "Use of Sentinel 2 – MSI for water quality monitoring at Alqueva reservoir, Portugal.", *Proceedings of the International Association of Hydrological Sciences*, 380, 73-79(2018). DOI: <https://doi.org/10.5194/piahs-380-73-2018>
- [27] G Elvidge, C. D., Baugh, K., Zhizhin, M., Hsu, F. C., & Ghosh, T., "VIIRS night-time lights. *International Journal of Remote Sensing*", 38(21), 5860-5879(2017). DOI: <https://doi.org/10.1080/01431161.2017.1342050>.
- [28] "Fastapi", <https://fastapi.tiangolo.com/>
- [29] "Sentinelsat", 25 July 2016, <https://pypi.org/project/sentinelsat/0.6.5/>
- [30] "Snappy toolkit", <http://google.github.io/snappy/>
- [31] Chen, K., Wang, J., Pang, J., Cao, Y., Xiong, Y., Li, X., ... & Lin, D., "MMDetection: Open mmlab detection toolbox and benchmark". arXiv preprint arXiv:1906.07155 (2019). DOI: <https://doi.org/10.48550/arXiv.1906.07155>
- [32] "Streamlit", <https://streamlit.io/>
- [33] "Plotly", <https://plotly.com/>
- [34] "LeafletJS", <https://leafletjs.com/>
- [35] Baldini, I., Castro, P., Chang, K., et al., "Serverless Computing: Current Trends and Open Problems," arXiv:1706.03178,(2017). DOI: <https://doi.org/10.48550/arXiv.1706.03178>
- [36] Eivy, A., "Be Wary of the Economics of 'Serverless' Cloud Computing," *IEEE Cloud Computing* 4(3), 6-12 (2017). DOI: 10.1109/MCC.2017.32
- [37] John, A., Ausmees, K., Muenzen, K., Kuhn, C., and Tan, A., "SWEEP: Accelerating Scientific Research Through Scalable Serverless Workflows," *Proceedings of the 12th IEEE/ACM International Conference on Utility and Cloud Computing Companion (UCC '19 Companion)*, 43-50(2019). DOI: <https://doi.org/10.1145/3368235.3368839>
- [38] Malawski, M., Gajek, A., Zima, A., Balis, B., and Figiela, K., "Serverless execution of scientific workflows: Experiments with HyperFlow, AWS Lambda and Google Cloud Functions," *Future Generation Computer Systems* 110, 502-514(2020). DOI: 10.1016/j.future.2017.10.029
- [39] Burkat, K., Pawlik, M., Balis, B., et al., "Serverless Containers – Rising Viable Approach to Scientific Workflows," 2021 IEEE 17th International Conference on eScience (eScience), 40-49(2020). DOI: 10.1109/eScience51609.2021.00014
- [40] Eismann, S., Scheuner, J., van Eyk, E., et al., "A Review of Serverless Use Cases and their Characteristics," arXiv:2008.11110 [cs.SE](2021). DOI : <https://doi.org/10.48550/arXiv.2008.11110>
- [41] Raman, R., Livny, M., and Solomon, M. H., "Matchmaking: distributed resource management for high throughput computing," *Proceedings. The Seventh International Symposium on High Performance Distributed Computing (Cat. No.98TB100244)*, 140-146(1998). DOI: 10.1109/HPDC.1998.709966
- [42] Daw, N., Bellur, U., and Kulkarni, P., "Xanadu: Mitigating cascading cold starts in serverless function chain deployments," *Proceedings of the 21st International Middleware Conference (Middleware '20)(2020)*. DOI: <https://doi.org/10.1145/3423211.3425690>
- [43] Sabbioni, A., Rosa, L., Bujari, A., Foschini, L., and Corradi, A., "A Shared Memory Approach for Function Chaining in Serverless Platforms," 2021 IEEE Symposium on Computers and Communications (ISCC), 1-6(2021). DOI: 10.1109/ISCC53001.2021.9631385
- [44] Villamizar, M., Garces, O., Ochoa, L., et al., "Infrastructure Cost Comparison of Running Web Applications in the Cloud Using AWS Lambda and Monolithic and Microservice Architectures," 2016 16th IEEE/ACM International Symposium on Cluster, Cloud, and Grid Computing (CCGrid), 179-182(2016). DOI: 10.1109/CCGrid.2016.37

- [45] Fouladi, S., Wahby, R. S., Shacklett, B., et al., "Encoding, fast and slow: Low-latency video processing using thousands of tiny threads," Proceedings of the 14th USENIX Symposium on Networked Systems Design and Implementation (NSDI 17), 363-376(2017). DOI:10.5555/3154630.3154660
- [46] Vázquez-Poletti, J. L., and Llorente, I. M., "Serverless Computing: From Planet Mars to the Cloud," Computing in Science & Engineering 20(4), 73-79(2018). DOI: <https://doi.org/10.1109/MCSE.2018.2875315>
- [47] "Amazon S3 Pricing", <https://aws.amazon.com/s3/pricing/>
- [48] "Amazon EC2 On-Demand Pricing", <https://aws.amazon.com/ec2/pricing/on-demand/>
- [49] "AWS Lambda Pricing", <https://aws.amazon.com/lambda/pricing/>