ORCS for RACE – Object Recognition and Classification from Satellite

Andrea Cavallini, PMP, PMI-ACP
Project Manager | Technical Lead | Earth Observation and AI Specialist

Diego Moglioni
AI System Engineer

Living Planet Symposium 2022

Object Recognition and Classification from Satellite (ORCS)
Product features

- AI-based object detection application
- It can run either as a Python stand-alone application as well as integrated in a Jupyter Notebook
- Exposing OGC WPS 2.0 Standard interface enabling external calls via RESTful APIs
- Currently running within ESA Euro Data Cube (EDC) infrastructure supporting ESA / EC RACE Project (task: ships and parked airplanes detection)

Cargo vessels

Parked Airplanes

Parked Cars

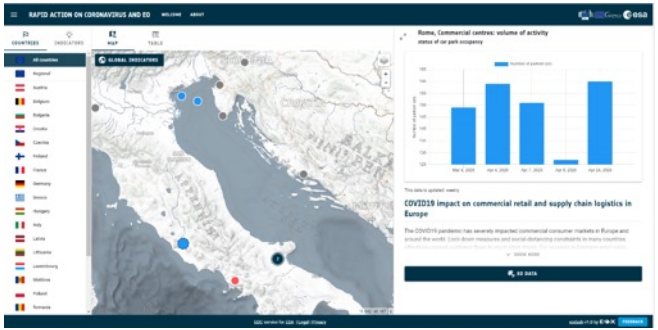
Parked Cars
(prototype only)

Early training using
Pleiades @ 50cm

ORCS for RACE

Rapid Action on COVID-19 and EO

- RACE is an open platform that uses EO satellite data and AI to measure the impact of the COVID-19 lockdown and to monitor post-lockdown recovery (joint initiative between ESA and EC).
- Web Dashboard exposing EO data monitoring and changes analysis
  - Economic operators
  - Environmental parameters
  - Human activities
- **ORCS provides automated information extraction** using Sentinel-2 data formerly performed by human operators
- Data covering an overall area of 4,500Km^2 (both ports and airports)
- Huge amount of data processed considering the area covered, Sentinel-2 10 m spatial resolution and 5 days revisiting interval

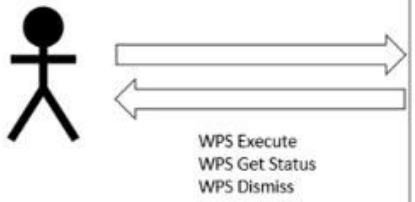


https://race.esa.int/

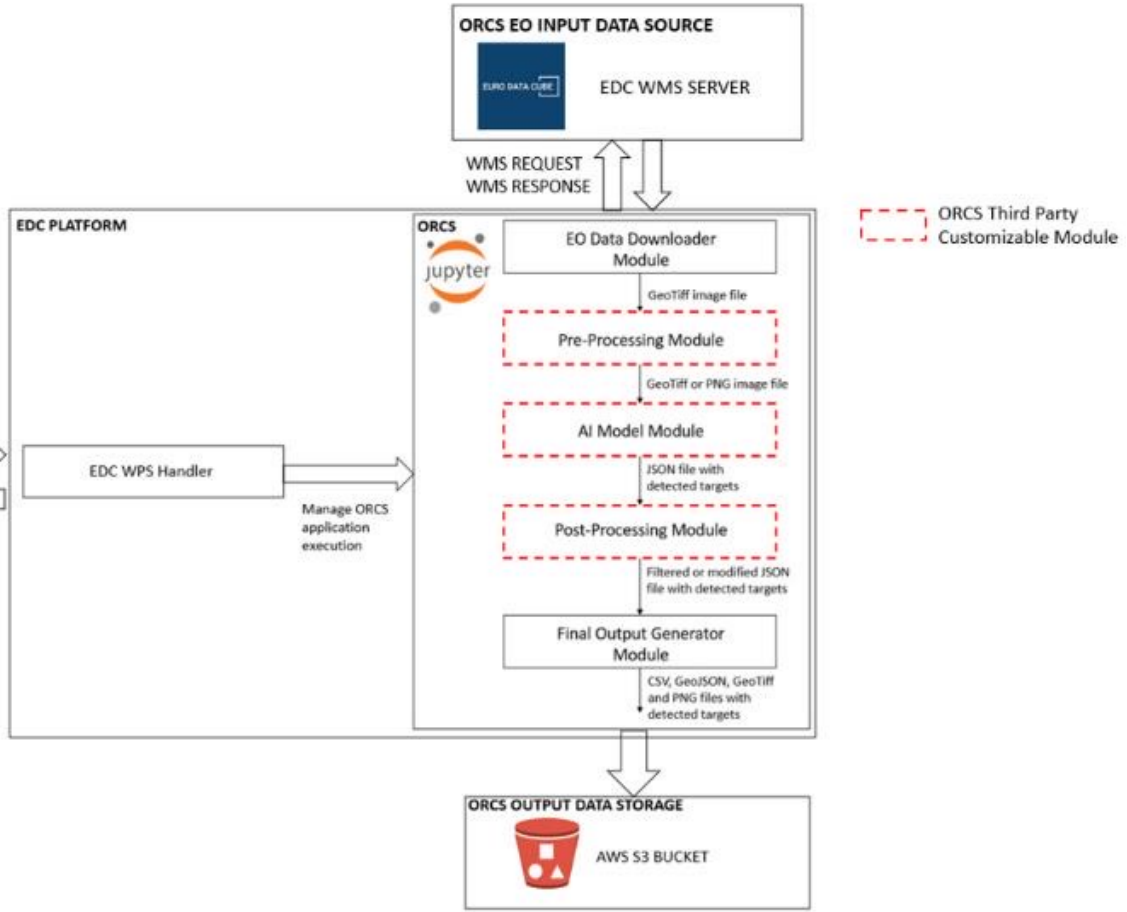
ORCS for RACE
Architecture overview

The overall architecture is designed to be

- **modular** to integrate additional modules that can be both new AI models and additional pre/post-processing modules
  - AI model → either a checkpoint file or any other model serialization file

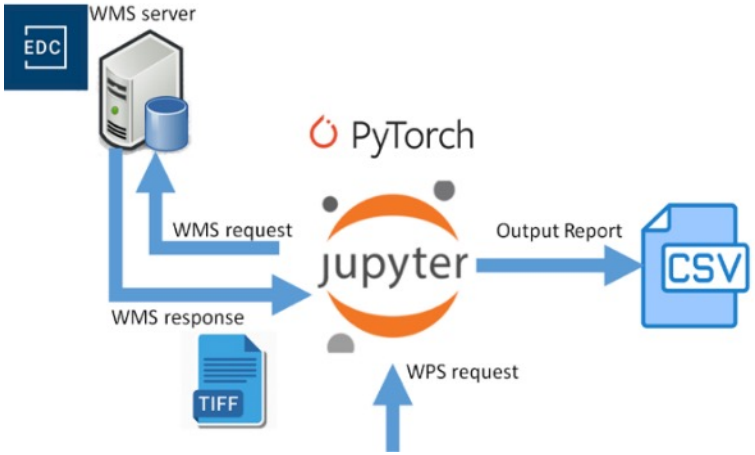


- a **framework** hosting and running external applications
  - JNs → e.g. flying airplanes detection



ORCS for RACE
System overview

- Specifications:
  - **Architecture:** Faster R-CNN (Fast R-CNN + RPN)
  - **Bands:** RGB
  - **Training:** ad-hoc training
  - **Platform:** running in EDC / OGC WPS interface
  - **Validation:** manual and automated tools
  - **Output:** Annotated GeoTIFF, GeoJSON, CSV
- Dataset:
  - **Input data:** Pleiades, PlanetScope, Sentinel-2
  - **Classes:** airplane, ship
  - **Preparation:** Semi-auto and manual annotation

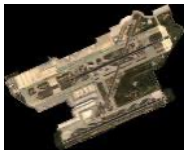


ORCS for RACE

Airplanes detection task: **AOI**

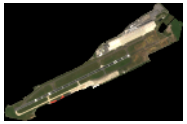

Amsterdam


Athens


Barcelona

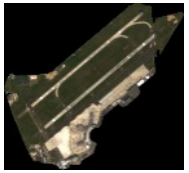

Berlin


Berlin


Paris


Copenhagen


Dublin


Dusseldorf


Frankfurt


Helsinki


Istanbul


Lisbon


London

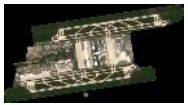

London


London


Malaga


Manchester


Milan


Munich


Oslo


Palma de Mallorca


Rome


Stockholm


Warsaw

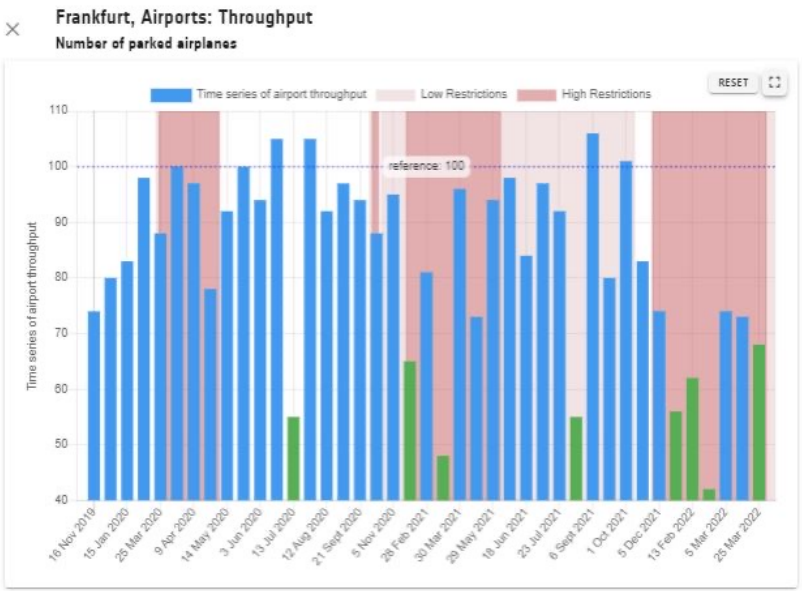

Wien


Zurich

ORCS for RACE

Airplanes detection task: **output example and dashboard visualization**

Frankfurt Airport



ORCS for RACE

Ships detection task: **AOI**

Dunkirk

Gdynia

Ghent

Suez

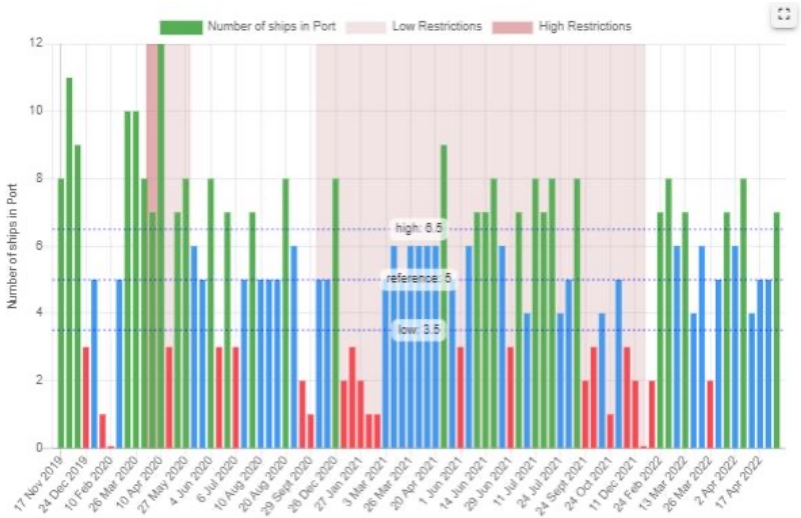
Genova

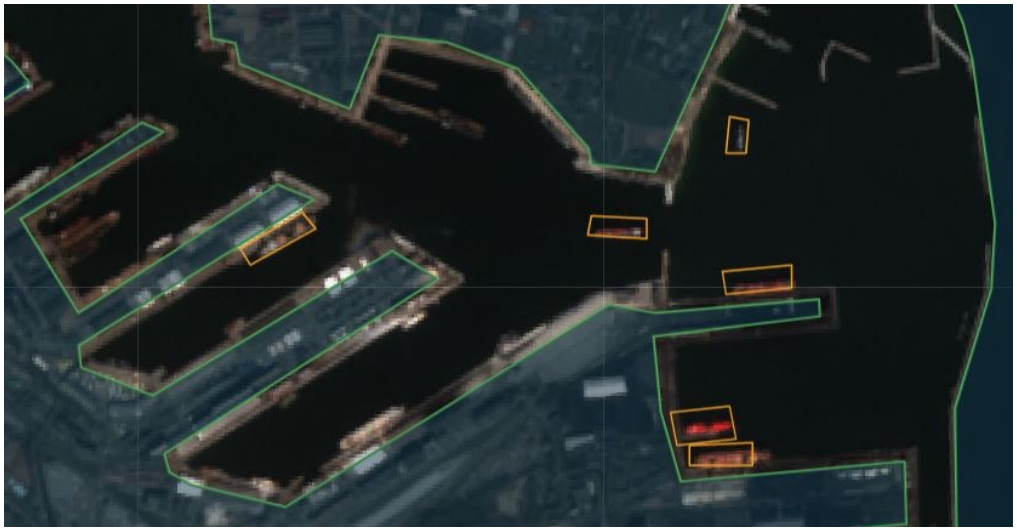
Hamburg

ORCS for RACE

Ships detection task: **output example and dashboard visualization**

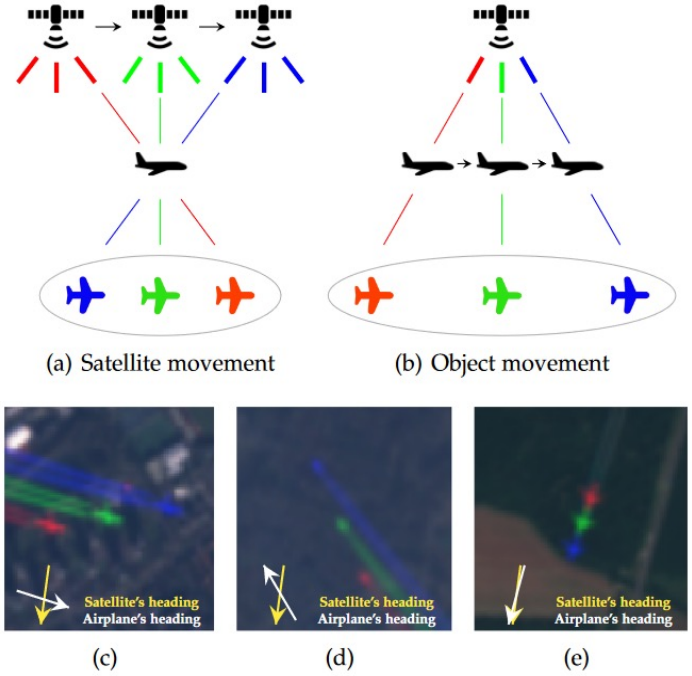
Gdansk, Gdnya Port





ORCS as a Framework
Flying airplanes detection

- Possibility to integrate external AI models / application / processing within *ORCS acting as a **framework** solution*
- Easing the integration within ESA – Copernicus Euro Data Cube environment and Copernicus Data Access / Sentinel Hub
- Currently used for identifying flying airplanes (AI model developed by IEEE members)
  - Flying airplanes instances are composed of three coloured blobs, each with the size of a parked airplane
  - A small delay between the acquisition of each colour band produces this effect for objects traveling at high speeds and/or high altitudes

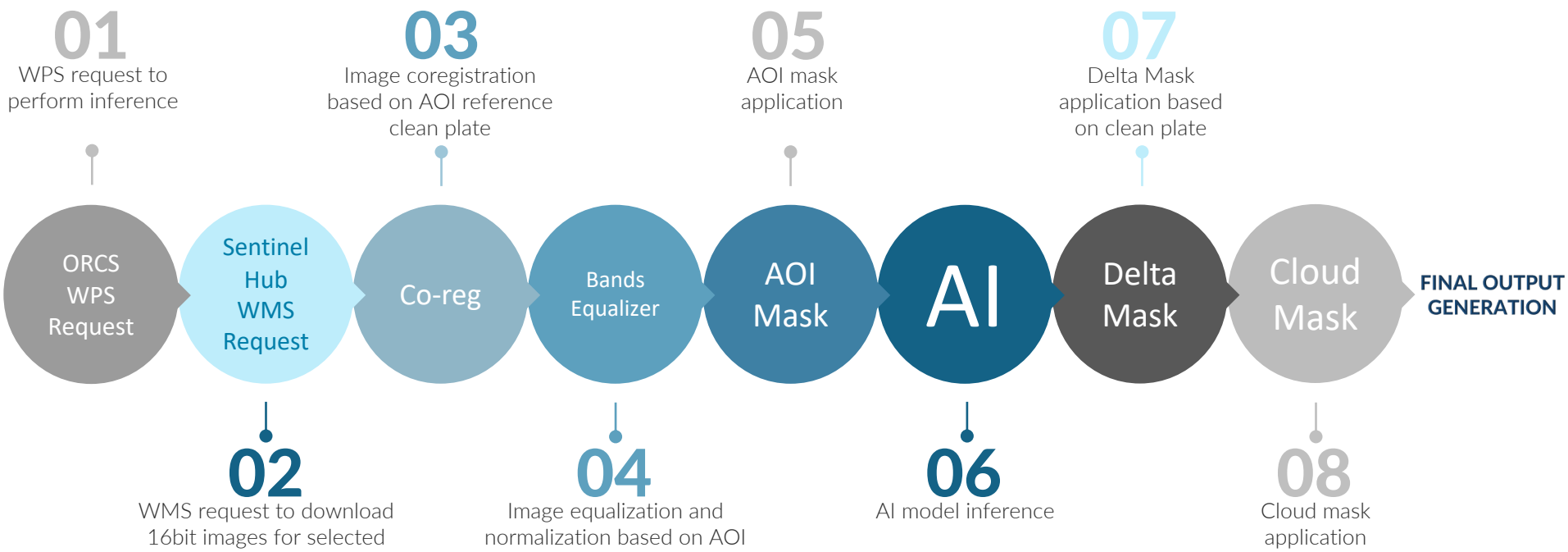


https://arxiv.org/pdf/2104.10345.pdf

ORCS workflow and AI architecture

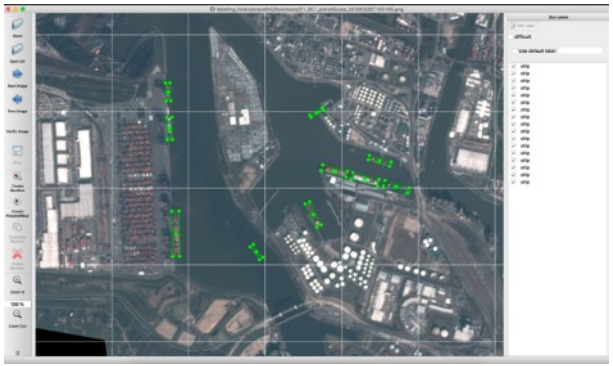
ORCS workflow and DL model

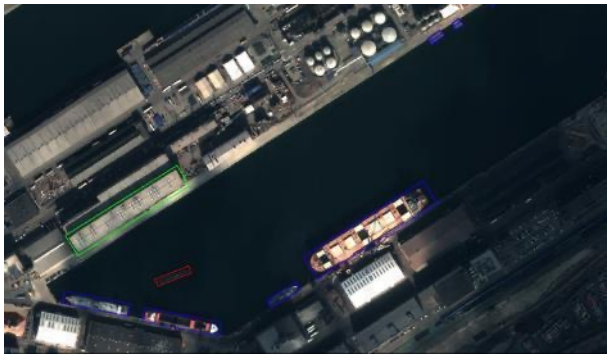
ORCS Workflow



Dataset preparation

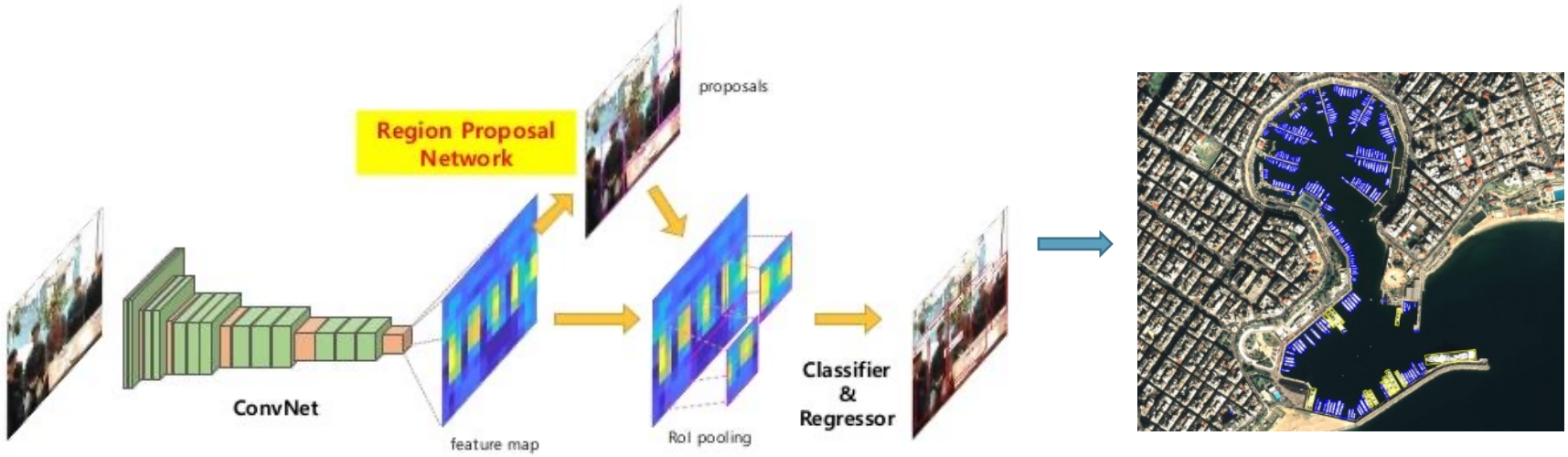
- Input data accessed via WMS
- Training datasets composed by > 500 annotated images (ships and airplanes)
  - Annotation files in Pascal VOC format
  - Manual annotation
  - Semi-Auto annotation





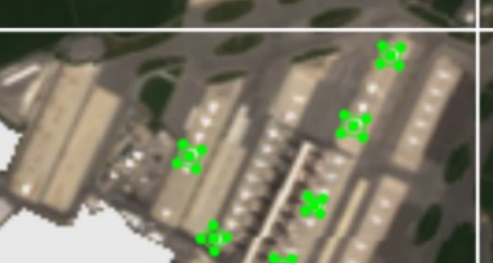
AI Model Architecture

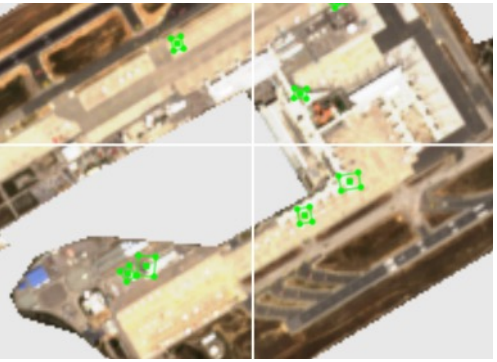
- Faster Region-based CNN and RPN (Regional Proposal Network) as backbone
- Classifier determines the probability of a proposal
- Regression regresses the coordinates of the proposals.

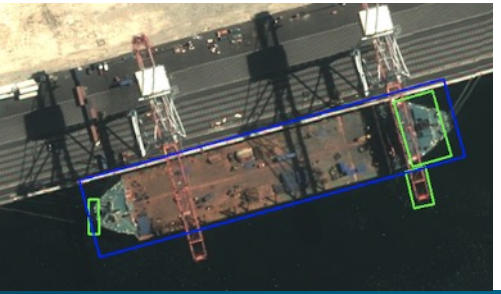


Challenges

Main challenges and performance improvements







Challenges

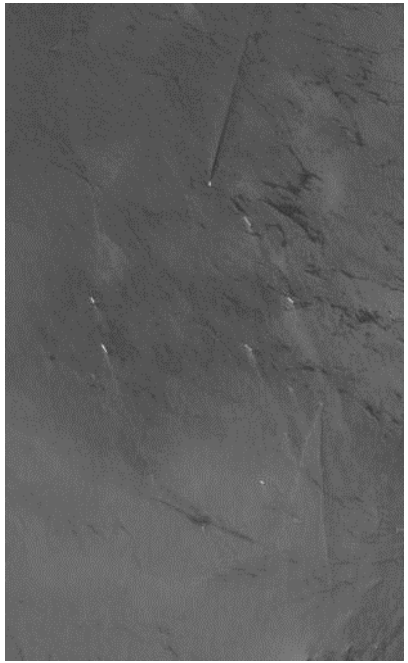
- Sentinel-2 Spatial Resolution of 10 m challenging for small airplanes or ships
- Different lighting conditions from acquisition to acquisition
- False positives on fixed structures, mainly in case of ports where some roofs with chimneys are sometimes recognized as ships.

Clean reference plates

- Creation of a plate for each AOI with the static elements of the acquisition
- Statistical approach (e.g., sigma, median, gaussian) to find the “true” and unvaried constant pixels
- Generated as multitemporal staking over a time series for each AOI
- It is used for several purposes:
  - For **Bands equalization** to overcome changing lighting conditions affecting also colors and saturation of target RGB image
  - For **delta-mask** application aimed to detect only the areas and features changed during the time
  - For **co-registration** task needed to apply correctly the **delta mask**

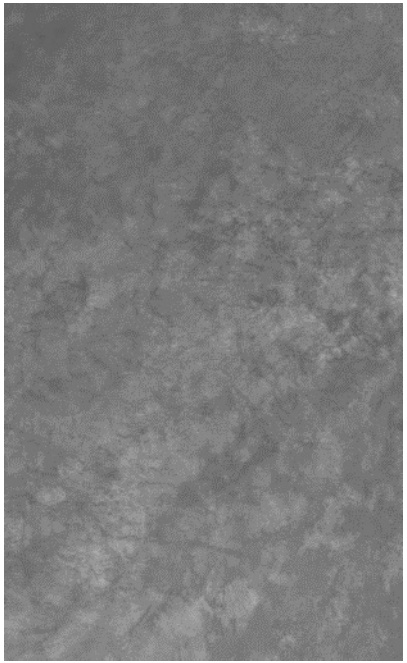
Delta Mask for ships sample

Validation image (grayscale)



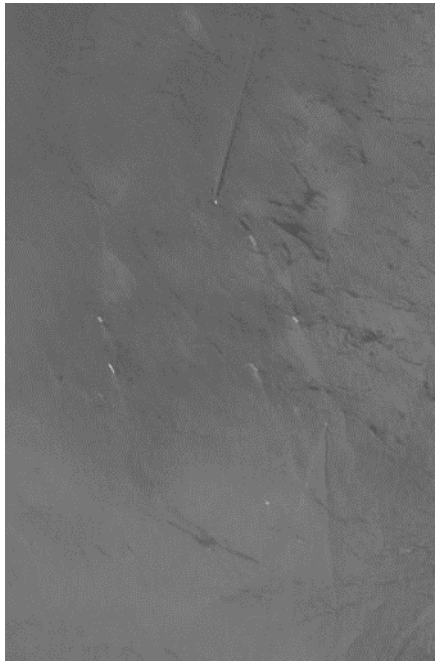
14-05-2017 08:38:35

Clean plate (grayscale)

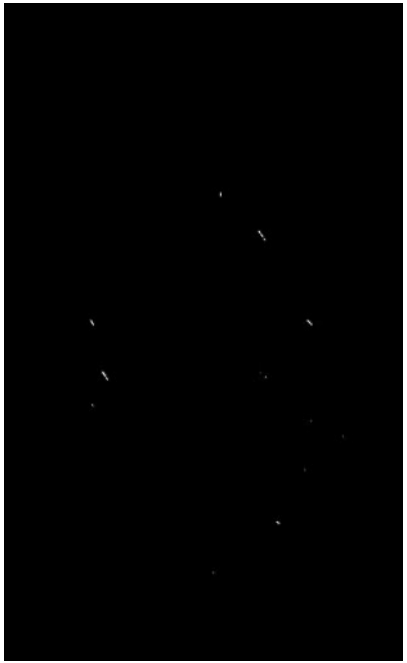


Median of 28 images

Delta (grayscale)



Delta (bitwise mask)



Post-processing
Cloud detection module

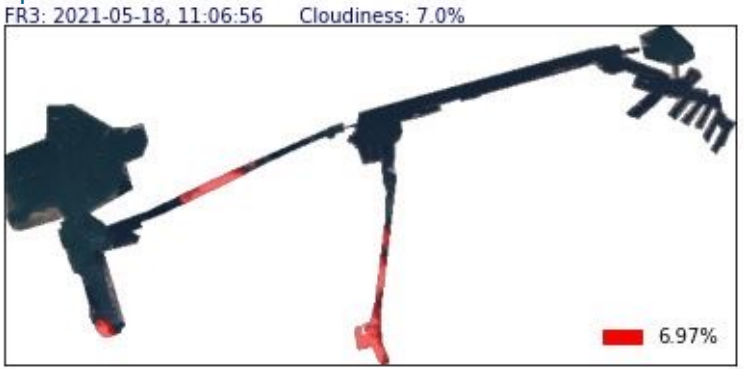
FEATURES

- Classification using single-scene pixel-based cloud detector (Sentinel Hub's s2cloudless)
- Performing cloud detection over AoI
- Calculating percentage of clouds (PoC) and filtering out cloudy images above a specified cloudiness threshold
- Removing cloud-covered detected objects

Clouds detection on harbours
Examples

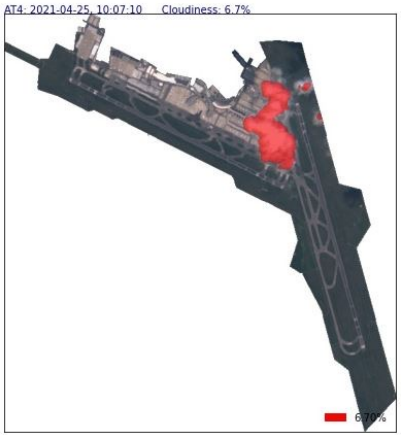
Port of Genova

Port of Dunkirque

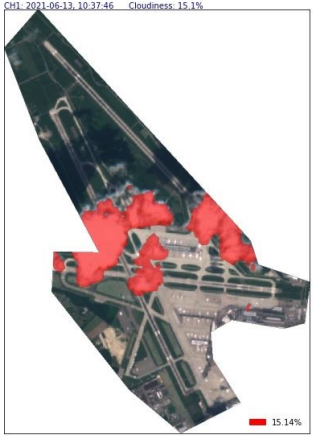


Clouds detection on airports
Examples

Vienna Airport



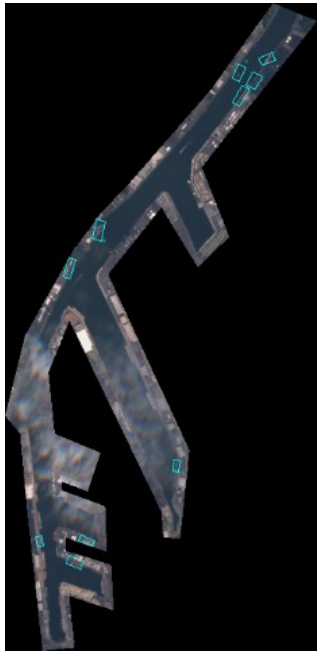
Zurich Airport



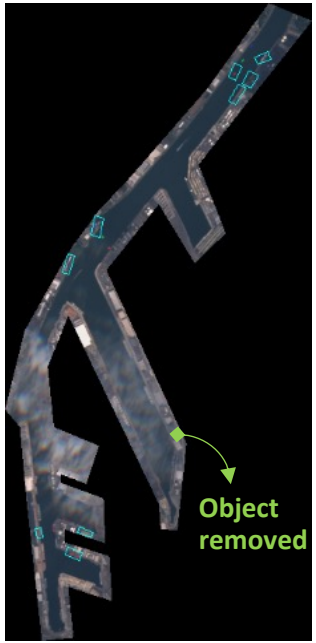
Clouds detection: removing cloud-covered objects

Example Ghent Zeehaven

Detection output



Cloud detection module output



Questions?

ORCS for RACE – Object Recognition and Classification from Satellite

Contact information: a.cavallini@rheagroup.com