Southern Gulf NRM

Satellite Based Water Analysis and Drought Planning

March 2025

Source: Ben Jarihani, 2025

Centre for Tropical Water & Aquatic Ecosystem Research (TropWATER) James Cook University

Townsville Phone: (07) 4781 4262 Email:

TropWATER@jcu.edu.au

Web: www.jcu.edu.au/tropwater/

The report may be cited as

Jarihani, B., Koci, J., Moore, K., Yoshida, M., Phelps, D., Felderhof, C., Gardner, L., Wharton, P., & Rodriguez Cubillo, D. (2025). *Satellite-based water analysis and drought planning*. Tropical North Queensland Drought Resilience Adoption and Innovation Hub, Townsville, Queensland.

Contacts

For more information contact: Dr Ben Jarihani, ben.jarihani@jcu.edu.au

Southern Gulf NRM

TNQ Drought Hub

TropWATER Centre for Tropical Water and Aquatic Ecosystem Research

JCU

JAMES COOK UNIVERSITY AUSTRALIA

ACKNOWLEDGMENTS

Sincere thanks to all landholders who generously allowed field surveys to be conducted on their properties or on adjacent common land. We are grateful to Southern Gulf NRM for facilitating engagement with landholders and supporting extension and communication activities throughout the project. This project was funded by the Tropical North Queensland Drought Resilience Adoption and Innovation Hub, through the Australian Government's Future Drought Fund. We extend our special thanks to Keleisha Moore for her valuable contribution to satellite data analysis, and to Momoe Yoshida for her role in developing the water body mapping app within the Google Earth Engine (GEE) platform. Their expertise was instrumental in delivering the technical components of this project.

We gratefully acknowledge the use of Landsat and Sentinel satellite imagery, made available through the U.S. Geological Survey and the European Space Agency, respectively. We also thank Planet Labs for providing high-resolution imagery that supported key components of this project. Special thanks to Geoscience Australia for access to the Water Observations from Space (WOfS) dataset, which contributed to water body validation and analysis. The project benefited significantly from the computational resources and cloud-based analysis capabilities of Google Earth Engine (GEE). These open-access data and tools were essential in enabling robust, scalable, and timely water monitoring to support drought resilience planning.

Table of Contents

EXECUTIVE SUMMARY

This project aimed to improve drought resilience and grazing land management by using satellite-based technologies to map and monitor small water bodies and farm dams across selected regions of the Southern Gulf. A partnership between Southern Gulf NRM (SGNRM) and James Cook University (JCU), the project delivered applied research, tool development, and stakeholder engagement to support data-driven decision-making for landholders and regional planners.

Six key activities were implemented:

1. **Development of mapping algorithms and methods** using satellite data from Landsat, Sentinel, and Planet to assess weekly changes in water body extent and dynamics. This delivered time series datasets, mapping tools, and interpretation products to support grazing land planning.
2. **Assessment of shrinkage and expansion of water bodies** over time, with analysis of their water-holding capacity to understand seasonal availability. Fifty selected water bodies were analysed to determine trends relevant to stock water management.
3. **Development of an automated water mapping app** using Google Earth Engine, enabling rapid and user-friendly monitoring by NRM groups and landholders.
4. **Validation and ground-truthing of remote sensing results** across various sites to ensure model accuracy and real-world applicability, involving extensive field checks and landholder engagement.
5. **Spatial analysis at the catchment scale** to evaluate water body distribution and density in key grazing areas. Although detailed paddock-scale analysis was limited by lack of fencing data, regional insights were successfully delivered.
6. **Communication and extension activities**, including a planned field day and a series of landholder meetings, were undertaken to share results and gather feedback. The project also contributed to a related Honours thesis, which will lead to a peer-reviewed publication.

The project resulted in the creation of a decision-support tool, improved water resource awareness, and insights into water access and grazing pressure. It also fostered knowledge sharing and built capacity among local landholders. Delivered under the Tropical North Queensland Drought Resilience Adoption and Innovation Hub and supported by the Australian Government's Future Drought Fund, this project reinforces the importance of science-based planning in building resilient agricultural systems in Northern Australia.

1 INTRODUCTION

Surface water availability is a critical factor for sustaining pastoral production, wildlife habitats, and irrigation across northern Australia's rangelands, which cover approximately 1.2 million square kilometres across Queensland, the Northern Territory, and Western Australia (Russell-Smith & Sangha, 2018). These landscapes support diverse ecosystem services and are integral to regional livelihoods (Brown & MacLeod, 2017). However, surface water is highly variable, with dams, creeks, and natural waterholes fluctuating dramatically in response to seasonal rainfall and climatic extremes (Jarihani et al., 2017). The availability and spatial distribution of these water sources directly influence grazing patterns, livestock welfare, biodiversity, and land management decisions. In particular, access to reliable water is essential during the dry season to reduce stress on cattle and avoid issues such as calf loss due to dehydration. Climate variability—driven by factors such as the El Niño Southern Oscillation (ENSO), Indian Ocean Dipole, and Madden-Julian Oscillation—further complicates water management, often resulting in prolonged dry spells or sudden flooding (Cobon et al., 2021; Eldridge & Beecham, 2018). In this context, monitoring surface water dynamics using remote sensing has become increasingly important for informed drought planning, property management, and resilience building in rangeland systems (Karfs et al., 2009; O'Reagain & Scanlan, 2013).

The Role of Small Water Bodies in Rangeland Systems

Despite their modest size, small water bodies (SWB)—ranging from 10 m^2 to 1 km^2—play a disproportionately large role in supporting biodiversity, ecological function, and agricultural productivity across rangeland landscapes (DeVries, 2017; Seelen et al., 2021). These features, which include ephemeral wetlands, ponds, and farm dams, are scattered across catchments and are often found near river systems, providing critical water retention during dry periods when streams may run dry (Littlefair & Matthews, 2024; Samways & Feest, 2020). In drought-prone regions, where ephemeral water bodies are more common than permanent water sources, agriculture relies heavily on constructed farm dams to support livestock, landscape health, and vegetation productivity—the latter being essential for grazing systems (Declerck & et al., 2006; Muller et al., 2007; O'Reagain & Scanlan, 2005). These water bodies not only provide drinking water for wildlife but also support local biodiversity and enhance ecological value (Borrelli et al., 2020; Seelen et al., 2021). However, existing satellite monitoring platforms such as MODIS (250 m) Landsat (30 m) often lack the spatial resolution to accurately detect or monitor smaller SWBs (Casal et al., 2021; Fisher & Acreman, 2016), resulting in significant gaps in water and land management practices. To ensure both ecological integrity and

agricultural resilience, there is a growing need for advanced monitoring tools specifically designed to detect and assess the condition of small-scale water systems in an improved spatial and temporal resolutions.

Surface Water Dynamics and Monitoring Challenges

The dynamic nature of small water bodies —particularly ephemeral systems—plays a crucial role in shaping biodiversity, ecological health, and agricultural sustainability. These water bodies expand during rainfall events, capturing runoff and sometimes causing flooding, which can have both positive and negative ecological impacts (Bolpagni et al., 2019; Shabarova & Pernthaler, 2021). In contrast, prolonged droughts and high evaporation reduce surface water availability, lower water tables, and increase pressure on both ecosystems and agricultural operations (Mayo & Loucks, 2021; Winter, 1999). Understanding and monitoring these fluctuations is essential for land-use planning, drought preparedness, and biodiversity conservation, especially in drought-prone regions (DeVries, 2017). While satellite platforms such as MODIS, Landsat and Sentinel have proven useful for detecting water extent, flood mapping, and wetland monitoring (Fisher & Acreman, 2016; Ogilvie & et al., 2018; Schwatke et al., 2020), current methods still struggle to capture volume changes and dynamics in smaller, irregularly shaped water bodies, especially those surrounded by vegetation. This gap in monitoring hinders effective management. Improving the accuracy and resolution of surface water monitoring tools is therefore essential to support agriculture, protect biodiversity, and mitigate environmental risks in a changing climate.

The primary objective of this project was to enhance grazing land management and drought resilience by improving the understanding and monitoring of small water bodies and farm dams using high-resolution satellite imagery. The project aimed to track the extent, dynamics, and availability of surface water over time, which is critical for supporting livestock, wildlife, and agricultural planning in drought-prone regions of Northern Australia. To achieve this, the project undertook several key activities: (1) development of algorithms for mapping water bodies and farm dams from satellite imagery; (2) assessment of water shrinkage, expansion, and holding capacity across selected sites; (3) development of a Google Earth Engine-based mapping tool for end-users; (4) field validation and ground-truthing of remote sensing outputs; (5) catchment-scale spatial analysis of water body distribution; and (6) communication of findings through landholder engagement, a planned field day, and contributions to a peer-reviewed scientific publication. These activities collectively contributed

to creating a practical, data-driven tool for landholders and NRM groups to support informed water and grazing management decisions.

2 METHODOLOGY

2.1 STUDY AREA

The study was conducted across two major river catchments in north-west Queensland, Australia: the Flinders and Gilbert River catchments, covering a combined area of approximately 190,000 km². The Flinders River, the longest river in Queensland and the sixth longest in Australia, originates in the Great Dividing Range, approximately 100 km north-east of Hughenden (Figure 1). These catchments support a range of economic activities, primarily beef cattle grazing, which is the dominant land use and a cornerstone of the regional economy. Additionally, the floodplains and river systems in these catchments offer significant potential for irrigated agriculture, particularly for crops such as cotton and grains, contributing to regional development and food production. The catchments also play a vital role in supporting biodiversity, Indigenous livelihoods, and ecosystem services, including water supply and landscape connectivity across the Gulf Savannah region. Their strategic location and natural resources make them central to Queensland's northern development and drought resilience planning.

The project focused on around 50 small water bodies, including farm dams, wetlands, and natural waterholes distributed across both catchments and surrounding areas. These water bodies were selected to capture a range of sizes, hydrological behaviours, and land-use settings relevant to grazing and drought resilience. In addition to sites across the Flinders and Gilbert systems, a number of selected water bodies located around Charters Towers were also included to test the satellite-based water monitoring approach in more intensively managed grazing properties. This spatial diversity provided a robust basis for assessing the performance and generalisability of the tools developed.

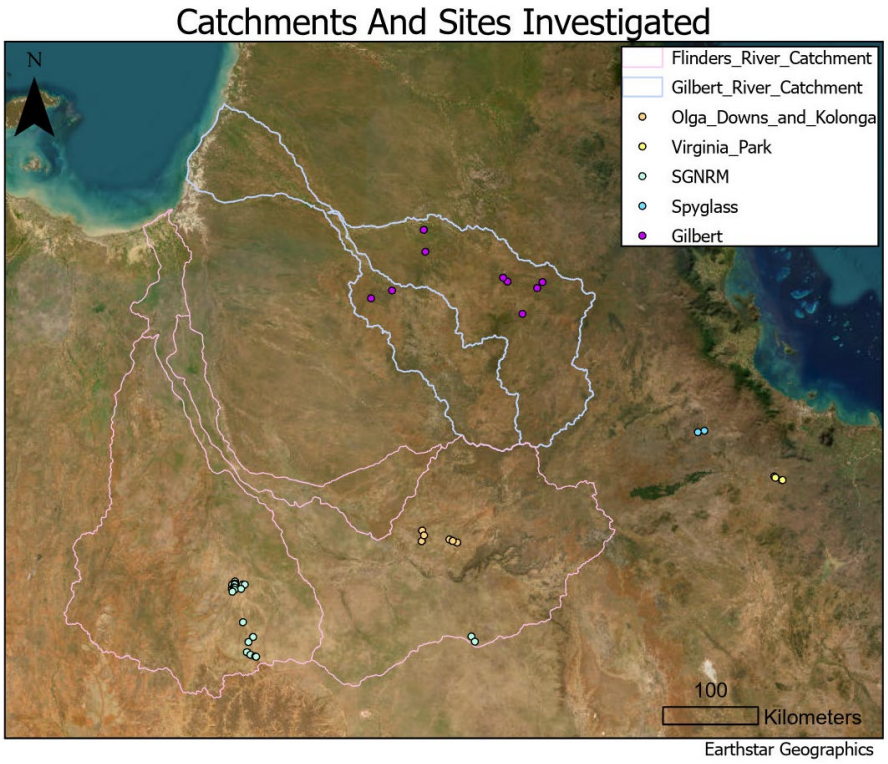


FIGURE 1. STUDY SITE, SELECTED WATER BODIES AND CATCHMENTS.

2.2 SATELLITE-BASED WATER EXTRACTION

To monitor the extent and dynamics of small water bodies, this project utilised optical satellite imagery from Landsat, Sentinel-2, and PlanetScope satellites. These platforms offer varying spatial and temporal resolutions, enabling both regional coverage and high-frequency monitoring of surface water. Water extraction from satellite imagery was primarily conducted using established spectral water indices, which highlight water features by leveraging the distinct reflectance properties of water compared to surrounding land and vegetation. The Normalized Difference Water Index (NDWI) and its modified version, MNDWI, were applied across the datasets to detect and delineate open water bodies (Table 1). In addition, high-resolution Planet imagery was used for manual digitisation of water extents, particularly in complex or vegetation-covered areas where automated methods were less reliable. These manually digitised polygons also served as validation references for testing the accuracy of the index-based extractions.

TABLE 1. SUMMARY OF WATER INDICES USED FOR SURFACE WATER EXTRACTION

Index	Formula	Bands Used	Purpose
NDWI	(Green – NIR) / (Green + NIR)	Green, Near-Infrared (NIR)	Enhances open water features; reduces noise from vegetation
MNDWI	(Green – SWIR) / (Green + SWIR)	Green, Shortwave Infrared (SWIR)	Better at removing built-up areas and improving water detection
AWEI	4*(Green – SWIR) – (0.25*NIR + 2.75*SWIR2)	Green, NIR, SWIR, SWIR2	Automated Water Extraction Index, useful in wetland areas

Note: Band availability varies between satellites. All indices were adjusted for platform-specific band configurations and spatial resolution.

For Sentinel-2, the Sentinel Dynamic Land Cover (DLC) product was used to isolate the water layer, which was accessed and processed using the Google Earth Engine (GEE) platform. A custom algorithm was developed in GEE to automate the extraction and analysis of water features across the study sites. For Landsat imagery, the Normalized Difference Water Index (NDWI) was applied to enhance the contrast between water and surrounding land surfaces, enabling accurate classification of water extent. For PlanetScope imagery and drone imagery, water bodies were manually digitised due to their high spatial resolution, which allowed for precise delineation, particularly in areas with complex shorelines or partial vegetation cover. These multiple approaches ensured flexibility in analysis and improved the robustness of water body detection across varying landscape conditions.

2.3 DRONE-BASED WATER MAPPING

In addition to satellite-based methods, drone imagery was employed to provide very high-resolution mapping of selected small water bodies. Drone surveys were conducted using RGB cameras, achieving spatial resolutions better than 5 cm, which enabled detailed observation of water boundaries, shoreline conditions, and surrounding land cover. This ultra-fine resolution allowed for the detection of small, shallow, or irregularly shaped water bodies that may not be fully captured by satellite imagery due to limitations in pixel size or interference from surrounding vegetation.

Drone-derived orthomosaics were processed using standard photogrammetry techniques and used to validate and compare against satellite-derived water extents, offering a benchmark for evaluating classification accuracy. These drone-based observations also helped improve our understanding of site-specific water body characteristics, such as partial vegetation cover, muddy edges, or disconnected sub-pools—factors that often challenge automated remote sensing approaches. Overall, drone imagery played a key role in ground-truthing and improving the reliability of satellite-derived water mapping products.

2.4 TEMPORAL ANALYSIS AND WATER AREA DYNAMICS ASSESSMENT

To assess the temporal dynamics of water bodies, a multi-scale approach was employed using satellite imagery across both catchment-scale and individual site-scale analyses.

At the catchment scale, we selected two representative images for each catchment—one from the wet season and one from the dry season—to extract and compare the spatial distribution and extent of water bodies. This seasonal comparison provided insight into the broader hydrological variability and water retention characteristics across the Flinders and Gilbert catchments.

At the site scale, monthly water area extraction was performed for each of the 50 selected water bodies using a consistent set of satellite images. This enabled a detailed analysis of water area fluctuations over time, capturing patterns of expansion and contraction in response to rainfall events, evaporation, and local catchment conditions. By analysing monthly water area changes, we were able to assess intra-annual

variability and better understand the persistence, reliability, and potential drought vulnerability of each water body.

This temporal assessment is crucial for informing grazing land planning, especially during the dry season, when water access becomes a limiting factor for livestock and land condition. The results also provide a foundation for evaluating the effectiveness of farm dams and natural waterholes in supporting drought resilience strategies.

2.5 ACCURACY ASSESSMENT

An accuracy assessment was conducted to evaluate the reliability of satellite-derived water body extents. The analysis included data from three satellite platforms—Landsat, Sentinel-2, and PlanetScope—applied across all 50 selected water bodies. For a subset of sites, drone imagery with a spatial resolution better than 5 cm was used as the ground truth reference, providing highly accurate water boundaries for comparison.

Where drone imagery was not available, we relied on high-resolution Google Earth imagery and/or PlanetScope satellite data (3–5 m resolution) as proxy references to assess the accuracy of water classification. These alternative basemaps were selected based on recency and visual clarity, particularly in dry-season conditions when water boundaries were clearly defined.

For each comparison, we calculated the percentage difference in detected water area between the satellite-derived classification and the reference data. Results were reported as either an overestimate or underestimate, depending on whether the satellite output exceeded or fell short of the reference water extent. This approach allowed us to quantify the positional and area-based discrepancies in water detection and better understand the performance of each satellite platform across various landscape conditions and vegetation types.

2.6 APP DEVELOPMENT IN GOOGLE EARTH ENGINE (GEE)

As part of this project, a user-friendly water mapping application was developed using the Google Earth Engine (GEE) platform. The app provides an interactive, web-based environment that enables users—such as landholders, NRM officers, and researchers—to easily explore and visualise surface water extent across selected regions.

The app allows users to select an area of interest on a map, choose a specific date, and then view water area data corresponding to that time and location. The goal of the app is to offer an accessible interface for non-technical users to engage with remote sensing data for practical water management and drought planning.

The current version of the app includes three water-related products:

1. Water Observations from Space (WOfS) – Derived from over 30 years of Landsat imagery, this layer provides information on the historical frequency and maximum extent of surface water for any given area. It helps users understand the long-term variability and reliability of water bodies.

2. Sentinel-2 Dynamic Land Cover Water Layer – Extracted from the national land cover product, this layer provides a recent classification of surface water, based on optical imagery and updated regularly to reflect seasonal conditions.
3. Sentinel-1 Radar-Based Water Detection – This layer is currently under development and uses synthetic aperture radar (SAR) data to detect water bodies regardless of cloud cover. While promising, this layer still requires further calibration and validation to improve accuracy and consistency across different land cover types and seasonal conditions.

The app represents a key output of the project, supporting the transition of satellite data into practical tools for decision-making. It can assist with planning water infrastructure, identifying high-risk areas during dry seasons, and promoting a better understanding of surface water dynamics across rangeland landscapes.

3 WATER MAPPING RESULTS

3.1 TEMPORAL ANALYSIS AND WATER AREA DYNAMICS ASSESSMENT

The temporal analysis of water body dynamics across the study sites revealed distinct seasonal patterns in surface water extent, closely aligned with rainfall and climatic conditions. As shown in Figure 2, water areas generally peak during the wet season (December to March), followed by a gradual decline through the dry season. Comparisons between satellite datasets indicated that Sentinel-2 tended to overestimate water extent, particularly in complex or vegetated environments, while PlanetScope imagery provided more accurate and consistent water area estimates due to its higher spatial resolution.

Notable differences between Sentinel and Planet detections were observed, especially during the wet season, as illustrated in Figure 2. These discrepancies are largely attributed to cloud cover, which can obscure Sentinel-2 optical imagery during peak rainfall months, leading to missed or misclassified water bodies. In contrast, Planet's daily revisit frequency allows for greater image selection flexibility, enabling users to capture cloud-free views more reliably for accurate water mapping. These results highlight the importance of satellite selection and temporal resolution when monitoring small and ephemeral water bodies, particularly in dynamic rangeland environments where timely water data is critical for land management.

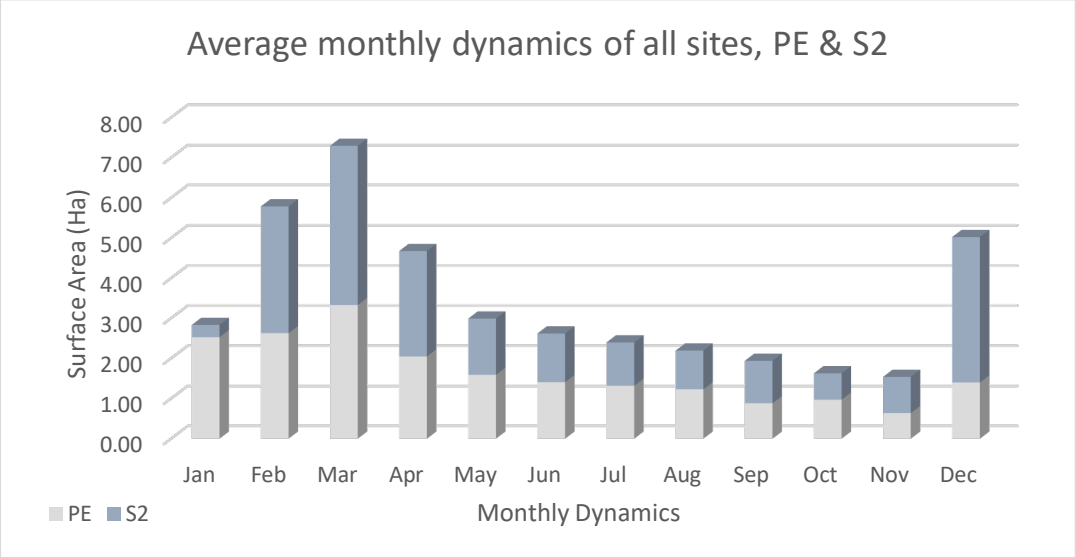


FIGURE 2: AVERAGE MONTHLY WATER AREA (IN HECTARES) ACROSS 50 SELECTED SITES OVER A 12-MONTH PERIOD, DERIVED FROM SENTINEL-2 AND PLANETSCOPE SATELLITE IMAGERY. THE CHART HIGHLIGHTS SEASONAL TRENDS IN WATER AVAILABILITY, WITH PEAK EXTENTS OBSERVED DURING THE WET SEASON (DECEMBER–MARCH) AND A GRADUAL DECLINE DURING THE DRY SEASON. DIFFERENCES BETWEEN THE TWO SATELLITE ESTIMATES REFLECT VARIATIONS IN SPATIAL RESOLUTION, REVISIT FREQUENCY, AND CLOUD COVER INTERFERENCE.

A comparative statistical analysis was conducted to evaluate the detection frequency of water presence between Sentinel-2 and PlanetScope satellites across the 50 selected water bodies. This analysis considered the number of times each satellite successfully detected water, no water, or failed to detect due to cloud cover or image unavailability. As shown in Figure 3, PlanetScope successfully detected water in 86% of observations, whereas Sentinel-2 detected water in only 63% of cases. This disparity is largely attributed to Planet's higher spatial resolution and daily revisit frequency, which allows more frequent acquisition of cloud-free imagery and improved detection of small and ephemeral water bodies.

In contrast, Sentinel-2 failed to detect water 32% of the time, often misclassifying small water bodies as dry due to its coarser resolution (10–20 m) and limitations in capturing smaller features. Furthermore, cloud cover during the wet season significantly affected Sentinel's ability to acquire usable imagery, highlighting a key challenge when relying solely on optical sensors with longer revisit cycles. Overall, the results demonstrate that PlanetScope offers superior reliability and consistency in water body detection, especially in dynamic and cloud-prone environments.

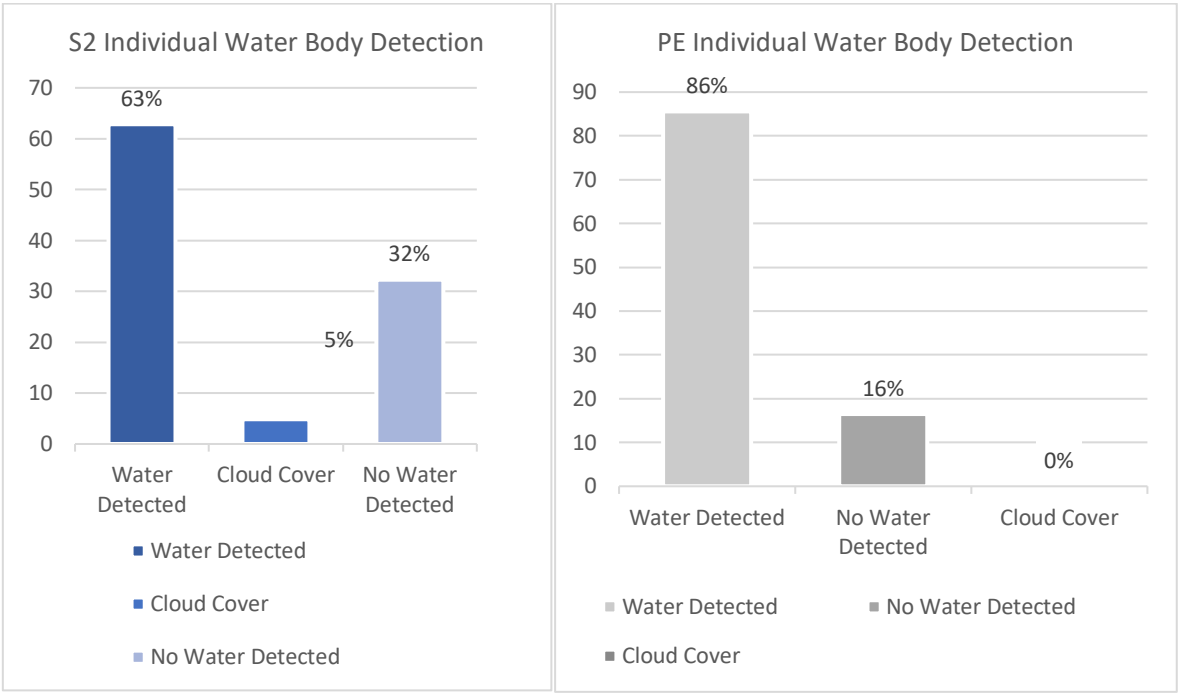


FIGURE 3: COMPARISON OF WATER DETECTION FREQUENCY BETWEEN SENTINEL-2 AND PLANETSCOPE ACROSS 50 SELECTED WATER BODIES. THE FIGURE SHOWS THE PERCENTAGE OF OBSERVATIONS WHERE EACH SATELLITE SUCCESSFULLY DETECTED WATER, DETECTED NO WATER, OR FAILED TO DETECT DUE TO CLOUD COVER.

3.2 COMPARISON OF SATELLITE-DERIVED WATER AREAS WITH DRONE IMAGERY

To evaluate the accuracy of satellite-derived water area estimates, a comparison was conducted using 15 water bodies that were surveyed with drones, providing high-resolution reference data (<5 cm). The analysis assessed the percentage difference between water areas extracted from PlanetScope and Sentinel-2 imagery against the drone-derived extents. The differences were calculated as overestimation or underestimation, with results shown in Figure 4, where negative values represent underestimation and positive values represent overestimation. The water bodies were sorted from smallest to largest along the horizontal axis to visualise size-dependent trends.

As observed in Figure 4, both Sentinel-2 and PlanetScope tended to underestimate water area for small water bodies, with Sentinel-2 showing greater underestimation. This is primarily due to coarser spatial resolution, which limits the satellites' ability to detect narrow or fragmented water features. In contrast, Planet's higher resolution allowed for more accurate detection but still missed some smaller water extents when compared with the drone baseline.

For larger water bodies, both satellites overestimated water areas relative to the drone surveys. This overestimation is likely caused by the inclusion of mixed pixels—particularly those containing water and vegetation or soil—being classified as water in the satellite-derived water detection algorithms. These findings highlight the importance of using high-resolution reference data when calibrating and validating satellite-based water mapping tools, and confirm that spatial resolution plays a critical role in the accuracy of water extent estimation, especially in small or ephemeral systems.

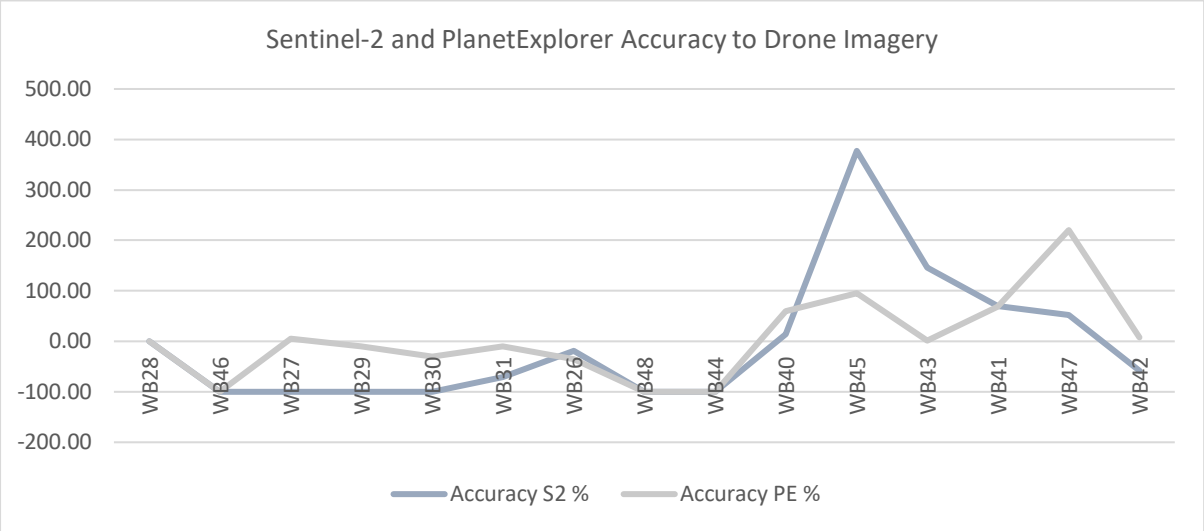


FIGURE 4: PERCENTAGE DIFFERENCE IN WATER AREA ESTIMATES FROM SENTINEL-2 AND PLANETSCOPE COMPARED TO DRONE-DERIVED WATER AREAS FOR 15 SELECTED WATER BODIES. NEGATIVE VALUES INDICATE UNDERESTIMATION, AND POSITIVE VALUES INDICATE OVERESTIMATION. WATER BODIES ARE ARRANGED FROM SMALLEST TO LARGEST (LEFT TO RIGHT) ON THE HORIZONTAL AXIS. THE FIGURE SHOWS THAT BOTH SATELLITES TEND TO UNDERESTIMATE SMALL WATER BODIES—MORE SIGNIFICANTLY IN THE CASE OF SENTINEL-2—WHILE OVERESTIMATING LARGER ONES DUE TO MIXED PIXEL CLASSIFICATION EFFECTS.

4 APP DEVELOPMENT RESULTS

A key outcome of this project was the development of a Google Earth Engine (GEE) web application designed to support water monitoring by providing an accessible, user-friendly interface for viewing surface water extent across the study area. The app was specifically developed to simplify the use of Earth Observation data for landholders, NRM groups, and other stakeholders, making it easier to visualise and interpret water availability without requiring technical GIS expertise.

The application features a simple graphical user interface (GUI), allowing users to interact with the map and perform custom queries. Users can select their area of interest using either a polygon drawing tool or a rectangular selection box. Once an area is selected, a date can be chosen via an integrated calendar. Users can then select from one of three available water products:

- Sentinel-2 Dynamic Land Cover water layer
- Sentinel-1 radar-based water detection layer
- Water Observations from Space (WOfS)

Upon selecting a dataset and a date, the app processes the request in real time and displays the estimated water inundation area within the selected region directly on the map interface. This allows users to compare water availability over time, assess trends, and gain insights into the water dynamics of their properties or catchments.

The tool offers a valuable step toward operationalising satellite data for drought preparedness and property-scale decision-making, bridging the gap between research and on-ground water management.

4.1 HOW TO USE THE WATER MAPPING APP

The Water Mapping App developed in Google Earth Engine (GEE) provides an intuitive, step-by-step process for users to extract surface water area within a region of interest. Below is a guide outlining how to use the app, accompanied by figures illustrating each step.

4.1.1 OVERVIEW OF THE APP INTERFACE

Upon launching the app, users are presented with a clean and interactive map interface with control options located on the left-hand panel.

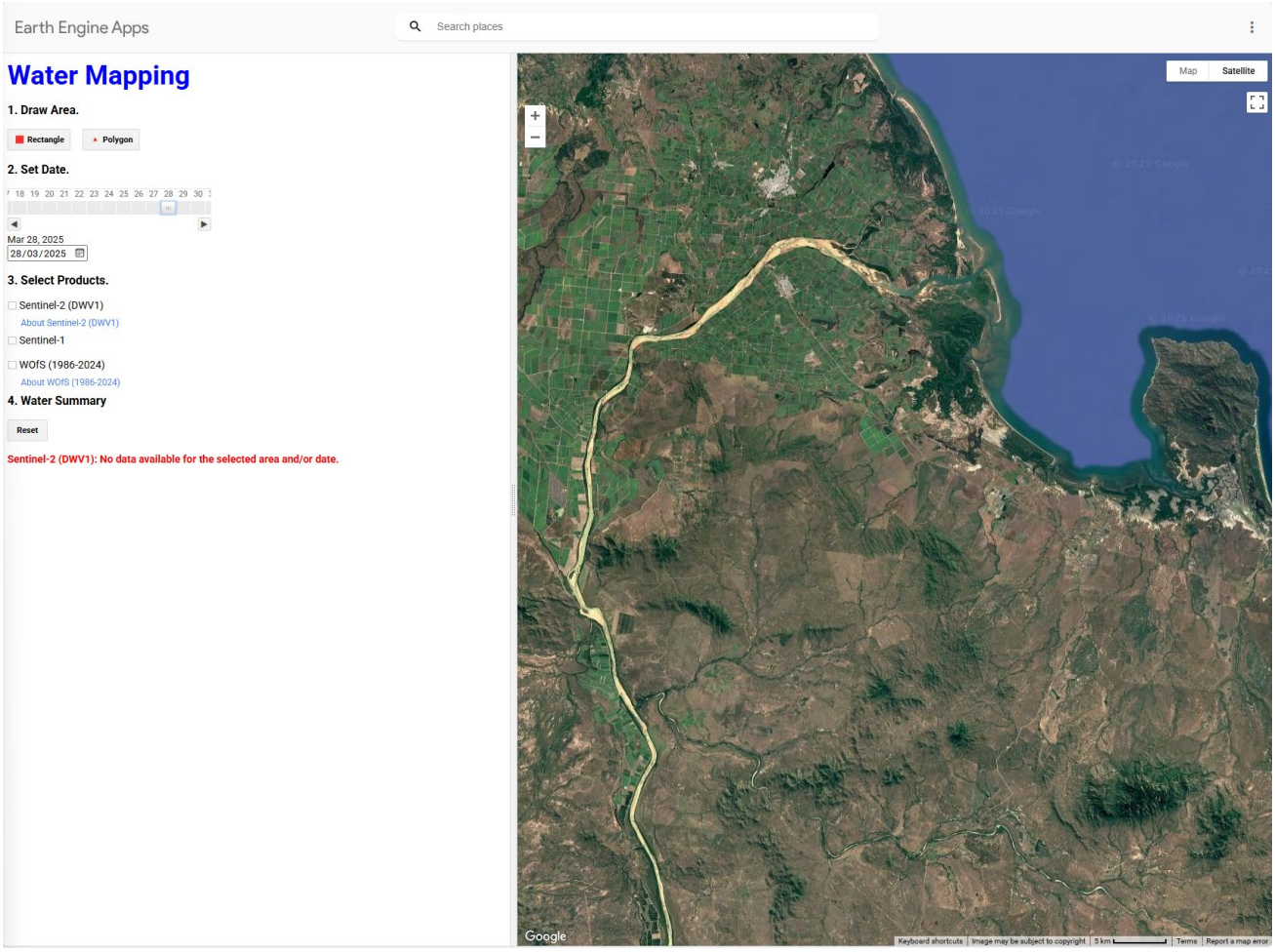


FIGURE 5. OVERVIEW OF THE GEE WATER MAPPING APP USER INTERFACE.

4.1.1.1 STEP 1: DRAW AREA OF INTEREST

Users can define the area for analysis using one of two selection tools:

- **Rectangular selection**
- **Polygon drawing**

These tools allow flexibility in defining the shape and size of the area of interest (AOI).

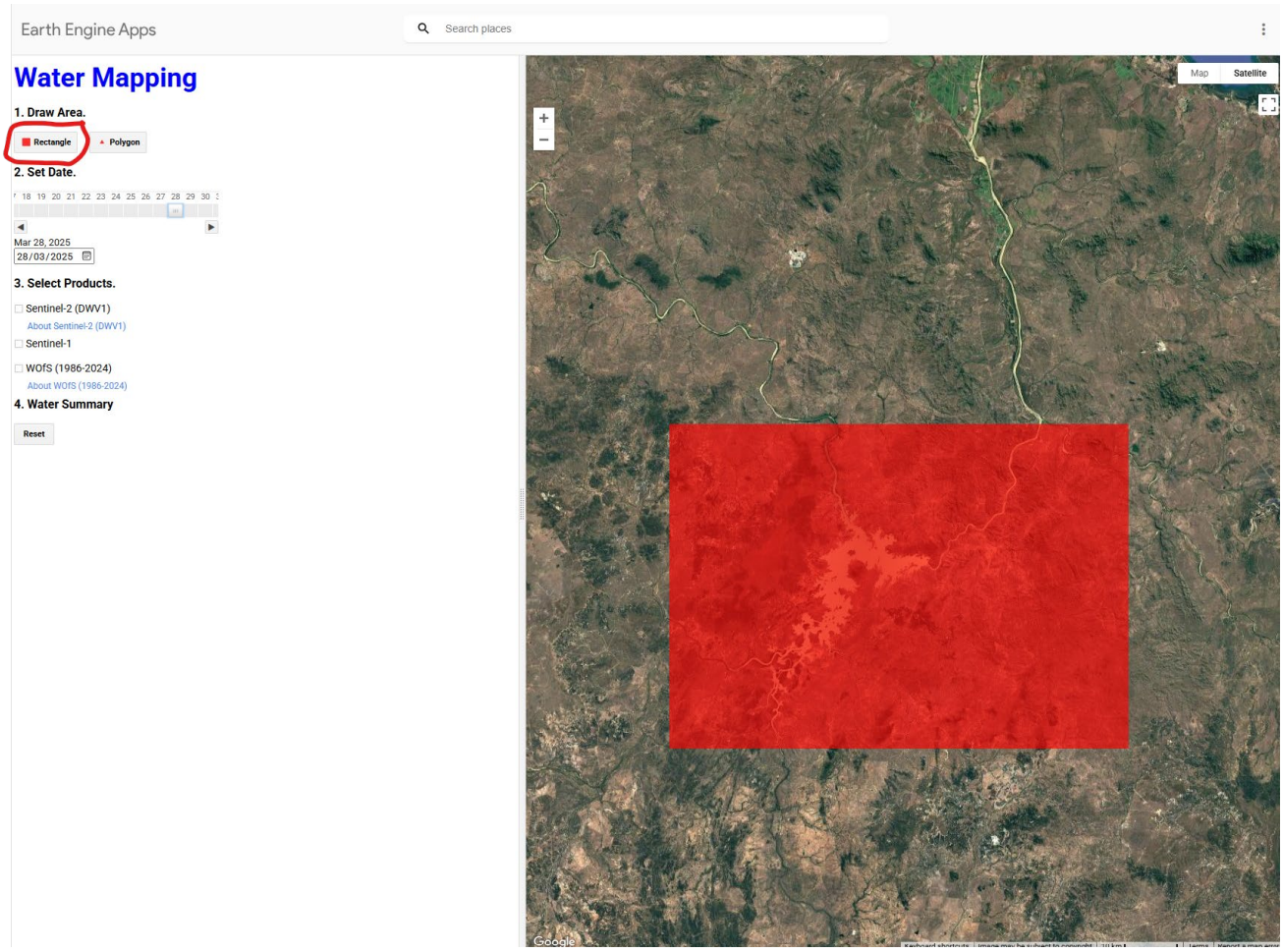


FIGURE 6. DRAWING A RECTANGULAR SELECTION BOX.

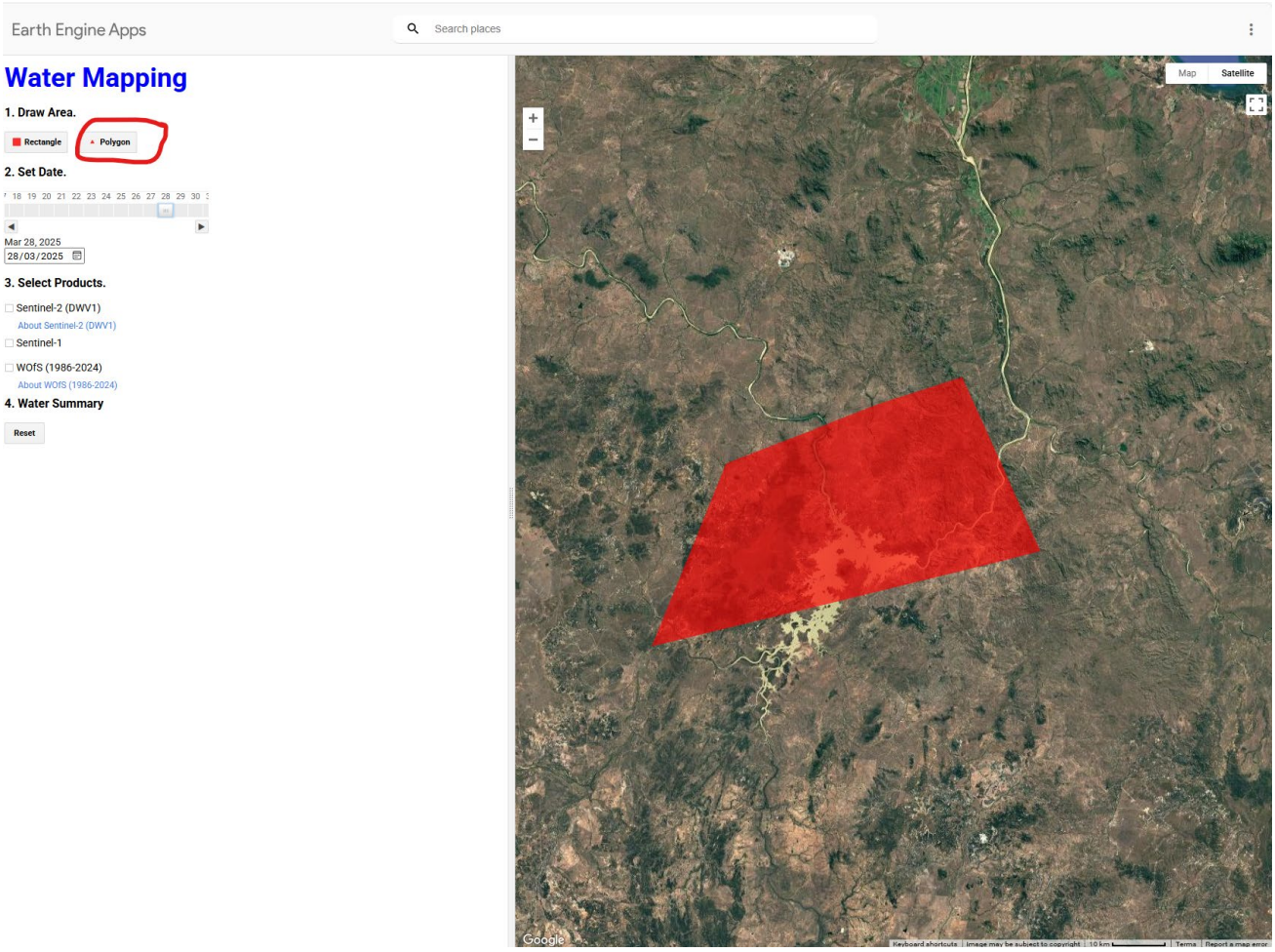


FIGURE 7. DRAWING A CUSTOM POLYGON SELECTION.

4.1.1.2 STEP 2: SELECT A DATE

Once an area is selected, users can specify the date for analysis using the **calendar tool**. This defines the imagery date from which water area will be calculated.

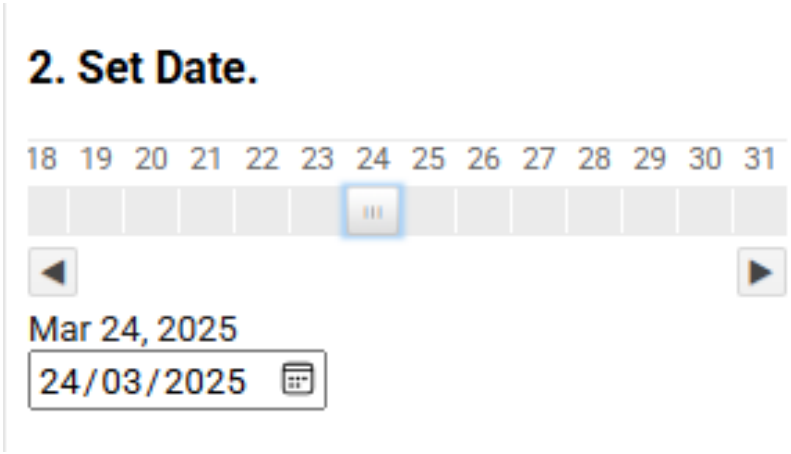


FIGURE 8. SELECTING A DATE USING THE CALENDAR WIDGET.

4.1.1.3 STEP 3: CHOOSE A WATER PRODUCT

Users then choose one of the following water mapping products to analyse the selected area:

- Sentinel-2 Dynamic Land Cover Water Layer
- Sentinel-1 Radar-Based Water Layer
- Water Observations from Space (WOfS)

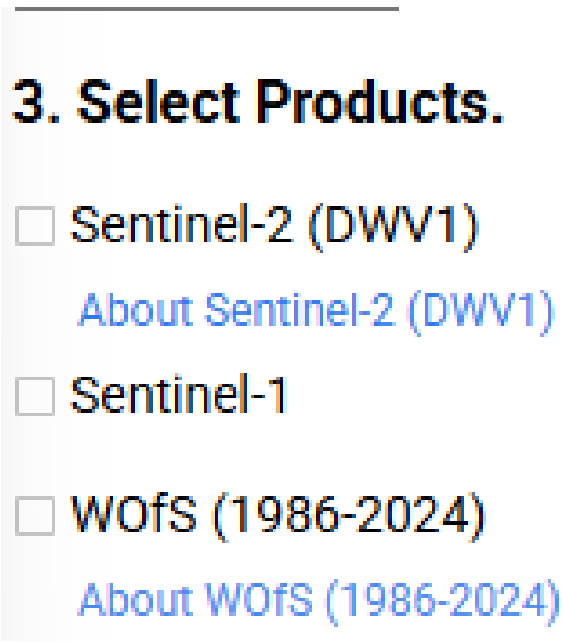


FIGURE 9. PRODUCT SELECTION MENU SHOWING AVAILABLE SATELLITE LAYERS.

4.1.1.4 STEP 4: VIEW RESULTS

Once a product and date are selected, the app will display the water area detected in the selected region. The result includes a visual overlay on the map and a numerical summary of the total water area in square meters.

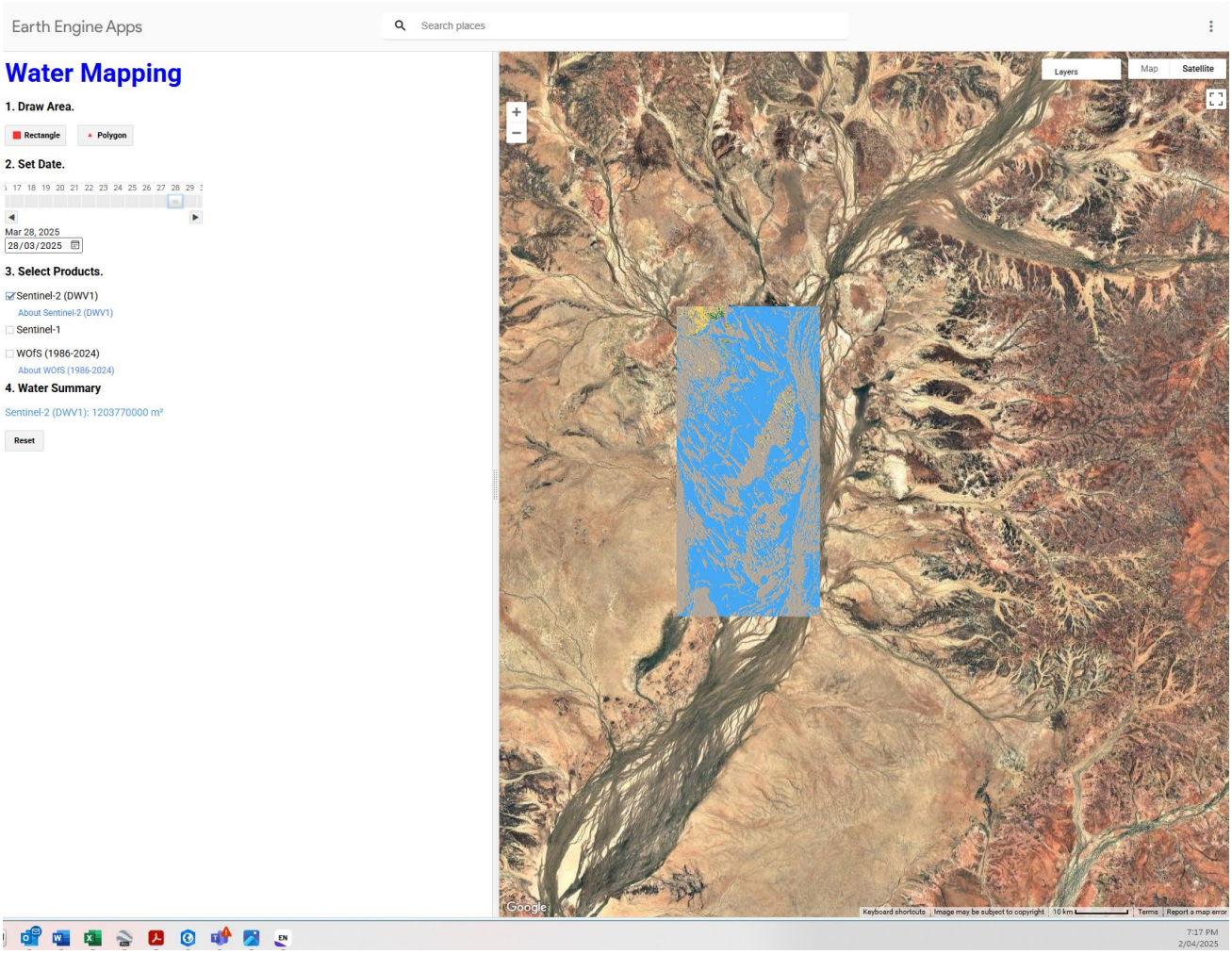


FIGURE 10. WATER AREA RESULT DISPLAYED ON THE MAP AND SUMMARY PANEL.

4.1.1.5 STEP 5: RESET ANALYSIS

To perform a new analysis or clear the current selection, users can click the Reset button, which clears the current AOI, product selection, and results.

5 DISCUSSION

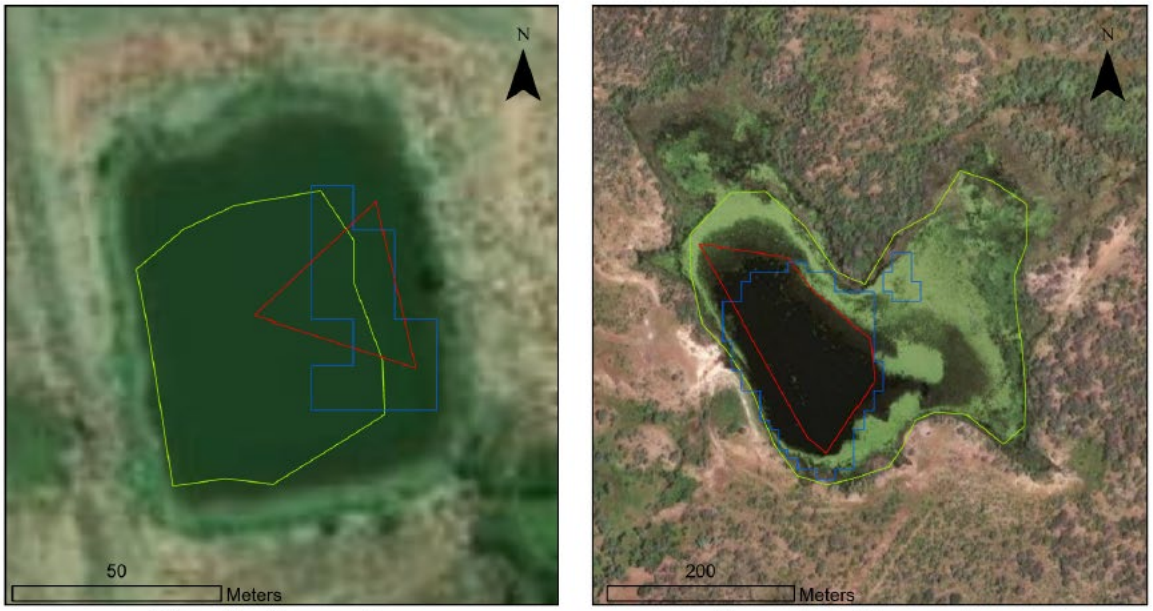
The accuracy of satellite-based water detection is influenced by a range of seasonal, environmental, and physical factors. During the wet season, high soil moisture surrounding water bodies can lead to overestimation of water extent, as moist or waterlogged soil may be misclassified as open water by spectral indices, particularly in medium-resolution imagery. Conversely, larger water bodies tend to yield more reliable detection results due to their size and reduced influence from surrounding land cover and edge effects.

In the dry season, there is typically greater spectral contrast between water and surrounding dry soil, which can enhance detection accuracy. However, smaller water bodies—especially shallow or ephemeral ones—are more likely to be misclassified or entirely missed, introducing greater uncertainty and error.

The presence of vegetation, both within the water body (e.g., aquatic or floating vegetation) and surrounding it (e.g., overhanging trees or dense riparian cover), further complicates classification. These features can obscure water surfaces or alter reflectance signatures, leading to underestimation of actual water extent. Additionally, the shape and geometry of water bodies also influence detection. Narrow, elongated rivers and waterholes are more prone to underestimation, particularly when overhanging vegetation or shadowing effects are present.

Other factors such as water depth and the colour or reflectivity of the underlying substrate (e.g., dark clay vs. pale sand) can also affect detection accuracy. Shallow water over light-coloured sand may reflect more light, making it harder to distinguish from dry areas, while deeper water typically presents a stronger, more consistent spectral signal. These challenges highlight the importance of using high-resolution imagery, incorporating seasonal calibration, and applying site-specific validation to improve the performance of remote sensing tools for surface water mapping in complex rangeland environments.

November Water Detection by L8, S2 & PE



Simple Geometry

Complex Geometry

Legend

L8_November S2_November PE_November

Source: Esri, Maxar, Earthstar Geographics, and the GIS User Community, Maxar

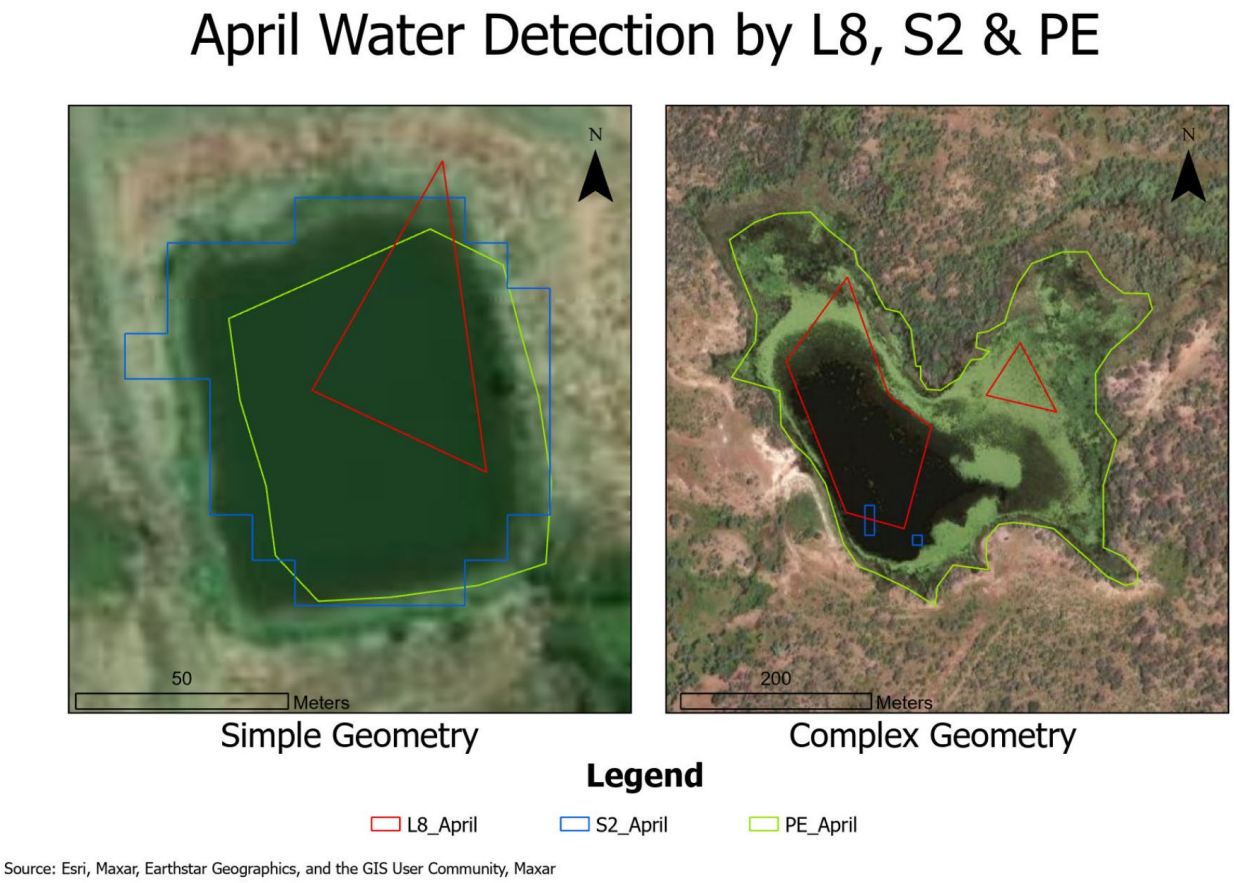


FIGURE 11: EXAMPLE OF TWO SELECTED WATER BODIES WITH SIMPLE AND COMPLEX GEOMETRY. EXTRACTED WATER AREA FROM LANDSAT8, SENTINEL2 AND PLANET SATELLITES ALSO PRESENTED FOR EACH WATER BODY.

6 FUTURE RESEARCH AND DEVELOPMENT

Building on the success of this project, several opportunities have been identified for future research and application development to enhance the monitoring and management of small water bodies in rangeland environments. A key area for advancement is the further development of the GEE-based Water Mapping App. Future work will focus on expanding the range of satellite products available in the app to improve its functionality and value for end-users. This includes integrating seasonal and near real-time datasets, and potentially incorporating outputs from hydrological models to provide predictive insights into water availability.

Another important direction is the refinement of the Sentinel-1 radar-based water detection algorithm, which offers a major advantage in cloud-prone or wet season conditions where optical imagery is often limited. Current development has shown promising results, but further calibration

and validation are required across different landscape types to ensure reliable detection of both ephemeral and permanent water bodies.

Additionally, there is significant potential to incorporate data from emerging altimetry satellites, such as the Surface Water and Ocean Topography (SWOT) mission, to estimate surface water elevation and volume. This would enable a more comprehensive understanding of water storage dynamics, particularly for deeper farm dams and natural lakes, complementing surface area mapping with depth and capacity estimates.

Together, these future developments will strengthen the capability of remote sensing tools to support drought preparedness, water resource planning, and land condition management, ultimately enhancing resilience in rangeland systems under increasing climate variability.

7 REFERENCES

Bolpagni, R., Bruno, F., & Bartoli, M. (2019). The hidden ecological role of small water bodies. *Ecohydrology & Hydrobiology*, *19*(3), 233-242.

Borrelli, P., Robinson, D. A., Panagos, P., Lugato, E., & Montanarella, L. (2020). Land use and climate change impacts on global soil erosion. *PNAS*, *117*(36), 21994-22001.

Brown, J., & MacLeod, N. (2017). Rangelands and ecosystem services: A framework for understanding and valuing the benefits of sustainable rangeland management. *The Rangeland Journal*, *39*(4), 313-320.

Casal, C., Díez, C., & Ferreiro, J. (2021). Limitations of medium-resolution satellite imagery. *Remote Sensing Applications*, *22*, 100493.

Cobon, D., Stone, R., Carter, J., Zhang, X., Rees, D., & Martin, P. (2021). Climate change and variability in grazing lands of northern Australia. *The Rangeland Journal*, *43*(2), 55-70.

Declerck, S. A. J., & et al. (2006). Multigroup biodiversity in shallow lakes. *Ecology*, *87*(7), 1687-1699.

DeVries, P. (2017). The role of small water bodies. *Wetlands Ecology and Management*, *25*(2), 123-134.

Eldridge, S., & Beecham, G. (2018). Understanding ENSO: What it means for Australia. *Bureau of Meteorology*.

Fisher, J., & Acreman, M. C. (2016). Wetland nutrient removal: a review. *Hydrology and Earth System Sciences*, *8*(4), 673-685.

Jarihani, B., Sidle, R. C., Bartley, R., Roth, C. H., & Wilkinson, S. N. (2017). Characterisation of Hydrological Response to Rainfall at Multi Spatio-Temporal Scales in Savannas of Semi-Arid Australia. *Water*, *9*(7), 540. https://www.mdpi.com/2073-4441/9/7/540

Karfs, R., Abbott, B., & Scarth, P. (2009). Land condition monitoring information for reef catchments: A new era. *QLD Department of Environment and Resource Management*.

Littlefair, C. J., & Matthews, T. J. (2024). Climate extremes and small freshwater bodies. *Global Change Biology*, *30*(1), e34512.

Mayo, A. L., & Loucks, D. P. (2021). Climate and water interactions. *Water Resources Research*, *57*(3), e2020WR028546.

Muller, B., Stadelmann, P., & Wehrli, B. (2007). Small-scale water bodies in agriculture. *Agricultural Water Management*, *92*(1–2), 1-10.

O'Reagain, P. J., & Scanlan, J. C. (2005). Sustainable grazing for tropical rangelands. *The Rangeland Journal*, *27*(1), 3-14.

O'Reagain, P. J., & Scanlan, J. C. (2013). Sustainable grazing management for temporal and spatial variability in north-eastern Australia. *The Rangeland Journal*, *35*(2), 131-142.

Ogilvie, A., & et al. (2018). Surface water monitoring using remote sensing. *Water Resources Management*, *32*(10), 3417-3433.

Russell-Smith, J., & Sangha, K. (2018). Sustainable land sector development in northern Australia: Indigenous rights, aspirations, and cultural responsibilities. *CRC Press*.

Samways, K. M., & Feest, A. (2020). Importance of ponds in agriculture. *Ecological Indicators*, *112*, 106052.

Schwatke, C., Dettmering, D., Bosch, W., & Seitz, F. (2020). DAHITI water levels from altimetry. *Hydrology and Earth System Sciences*, *24*(2), 891-908.

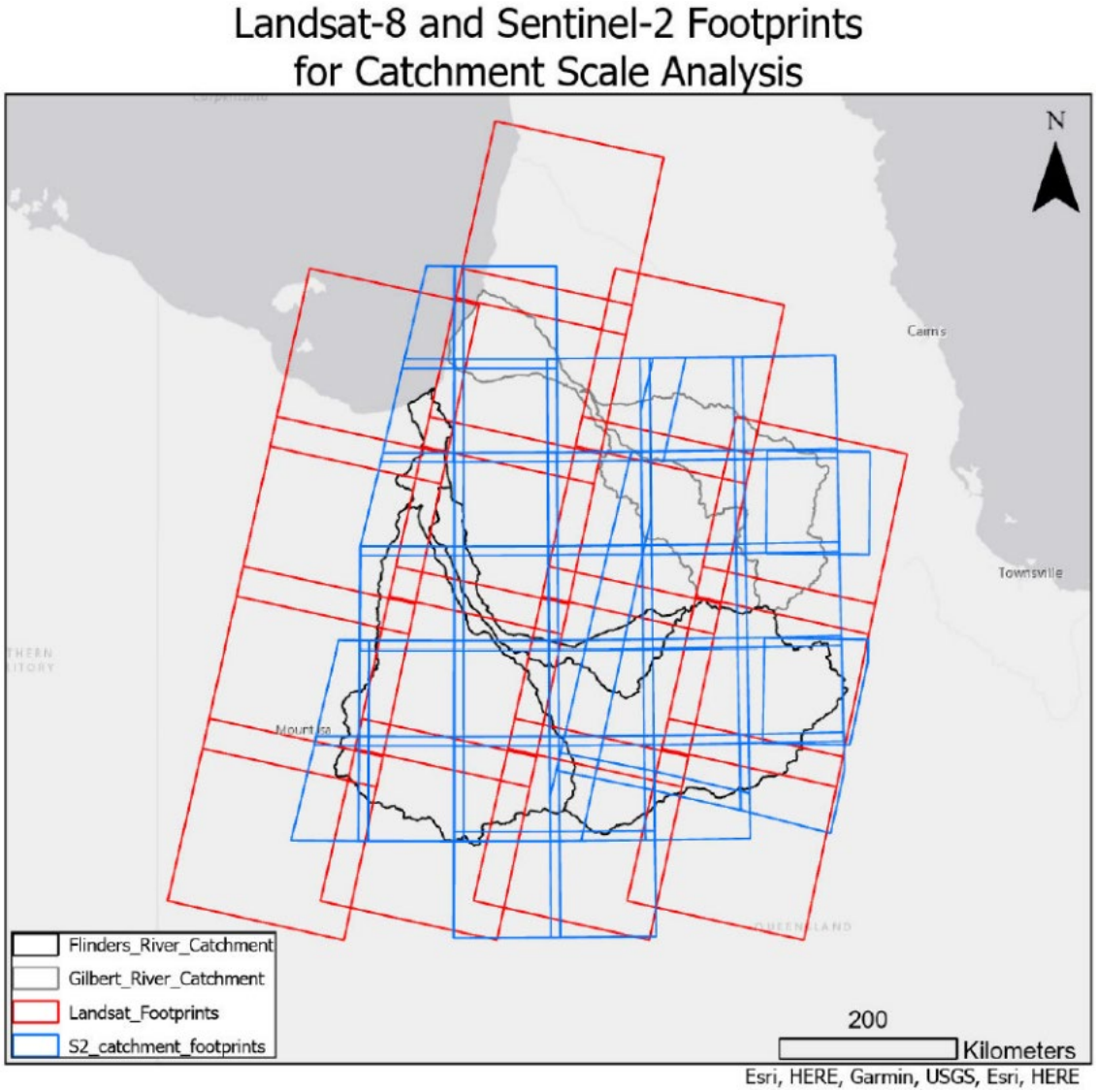
Seelen, L. M., Flaim, G., & Teurlincx, S. (2021). Small waterbodies for biodiversity. *Nature Sustainability*, *4*, 586-593.

Shabarova, T., & Pernthaler, J. (2021). Microbial responses to flood/drought. *Trends in Microbiology*, *29*(6), 464-476.

Winter, T. C. (1999). Hydrologic connectivity of water systems. *Hydrogeology Journal*, *7*(1), 28-45.

8 APPENDICES

APPENDIX 1: AVAILABLE LANDSAT-8 AND SENTINEL-2 FOOTPRINTS AT CATCHMENT SCALE FOR FLINDERS AND GILBERT



APPENDIX 2: SUMMARY OF KEY SATELLITE FOR SURFACE WATER.

Satellite/Product	Sensor Type	Spatial Resolution	Temporal Resolution	Water-Related Output	Remarks
Landsat (5, 7, 8, 9)	Optical	30 m	16 days	Surface water extent (NDWI, MNDWI), long-term analysis	Good for historical analysis (since 1980s)
Sentinel-2	Optical	10–20 m	5 days (with both satellites)	Surface water extent (NDWI, MNDWI), vegetation and land cover	Higher spatial and temporal resolution than Landsat
Sentinel-1	Radar (SAR)	10 m	6–12 days	Surface water detection (cloud-penetrating)	Useful in cloudy or rainy seasons, requires more calibration
PlanetScope	Optical	3–5 m	Daily	High-resolution surface water mapping	Commercial product, used for detailed/local-scale mapping
MODIS (Terra/Aqua)	Optical	250–500 m	Daily	Large-scale water extent, surface reflectance	Coarse resolution, more suitable for large water bodies
Water Observations from Space	Landsat-derived	30 m	Multi-year accumulation	Water presence frequency, maximum inundation	Useful for understanding historical patterns and variability
SWOT (Surface Water and Ocean Topography)	Radar Altimetry	~10–50 m	21 days (global repeat)	Surface water height, slope, and volume (future use)	Promising for water storage estimation, under ongoing validation

APPENDIX 3: SUMMARY OF KEY SURFACE WATER PRODUCTS.

Water Product:	Description:	Water extraction method/s:	Public Satellite/s:	Temporal & Spatial Resolutions:	Time period & global extent
Planet Explorer	Optical 3m pixel resolution satellite imagery that produces near-daily imagery.	Multiple satellites in a synchronous orbit (PlanetScope Doves)	PlanetScope, SkySat Sentinel-2 and Landsat 8	**T:** ranging from daily scenes to weekly, monthly and quarterly basemaps **S:** medium to high resolution	PlanetScope: global since 2016, 1 day revisit time SkySat: global since 2014 to present
DEA Water Observation Map	Provided by Geoscience Australia, uses Landsat 30m resolution to provide mainly historic satellite knowledge.	Water and non-water reflectance through vegetation & ground based mapping	Landsat 5, 7 and 8	**T:** Daily, monthly and yearly **S:** 30m	Last 50 years
Dynamic Landcover Map European Space Agency (ESA)	Universal Transverse Mercator (UTM) coordinate system aligned to the Sentinel-2 tiling grid.	Satellite in conjunction with Sentinel-2 tiling grid – UTM	PROBA satellite (PROBA-V) 100 m time-series	**T:** 2018 – 2020 **S:** Global coverage with 100m resolution	Global coverage Annually updated based on improved satellite imagery resolution changes per year (2020)
Hydrosheds	HydroSHEDS, HydroATLAS, HydroBASINS, HydroRIVERS, HydroLAKES, GLWD (Global Lakes and Wetlands Database), HydroWASTE, GloRiC (Global River Classification), HydroFALLS	SRTM (Shuttle Radar Topography Mission) DEM (digital elevation model)	DLR's twin TanDEM-X and TerraSAR-X satellites Landsat 5	**T:** 2007 and 2010 start **S:** 30m to 90m 3, 15, and 30 arc seconds	Global to regional applications
Water Product:	**Link and Description:**	**Water extraction method/s:**	**Satellite/s:**	**Temporal & Spatial Resolutions:**	**Time period & global extent**
DFO (Dartmouth	Global flooding observatory uses data from a range of sources to track flood events.	Data from NASA, Japanese Space Agency and ESA	This field depends on available data to map such flooding events. If data used		Established in 1993, depending on

Flood Observatory)			is record, it will be noted where relevant.		flooding events occurring
SWOT (Surface Water and Ocean Topography)	Produced by NASA/CNES to observe Surface Water and Ocean Topography.	Instruments: KaRIn, Poseidon-3C Altimeter (primary), AMR, DORIS, GPSP, LRA (additional) Relevant data sets: L2_HR_LakeSP L2_HR_LakeAvg L2_HR_PIXCVec	Depends on the dataset	**T:** 2024-01-25 to 2024-03-06 (current availability) **S:** Red Tile: 128km x 128km Green Tile: 64km x 64km	July 2023 start, pending fully operational launch?
OPERA DSWx (Dynamic Surface Water eXtent)	NASA - optical and SAR imagery for water dynamics on surface.	Combined end user satellite data: OPERA (Observational Products for End-Users from Remote Sensing Analysis)	Sentinel-1, NISAR, and the Harmonized Landsat Sentinel-2 (HLS) data	**T:** **S:** 30m pixels	Global extent Time Period:
ESA Lakes product	Dataset includes Lake Water Level, Lake water Extent, Lake surface water temperature, Lake Ice cover, Lake water-leaving reflectance and Lake Ice thickness.	Combination of multiple satellite instruments	TOPEX/Poseidon, Jason, ENVISAT, SARAL, Sentinel 2-3, Landsat 4, 5, 7 and 8, ERS-1, ERS-2, Terra/Aqua and Metop-A/B	**T:** ranging from daily up to 10 days **S:** ranging from 250m, 300m-1100m, 1km and 10km (depending on product input)	Global extent Time period: 1992, 1995, 2000, 2002 till 2022 (depending on product input) data set coverage.
Water Product:	**Link and Description:**	**Water extraction method/s:**	**Public Satellite/s:**	**Temporal & Spatial Resolutions:**	**Time period & global extent**
Dai Yamazaki Global Water Body Map	Last updated on 20/3/2018 Automated algorithm to process multi-temporal Landsat images from global land survey (GLS) database.	Global Land Survey database: 33,890 scenes from 4 GLS epochs	Landsat	**T:** multi-temporal **S:** 3 arc-seconds	Global extent

	Used 33,890 scenes from 4 GLS epochs in order to delineate a seamless water body map, without cloud and ice/snow gaps.	user: g3wbm pass: globalwater			
Impact Observatory (IO)	Complies annual data globally using 10m and 3m resolution imagery. Provides both free and paid products.	Nine land cover classes (Water, Trees, Rangeland, Flooded Vegetation, Crops, Built Area, Bare Ground, Snow/Ice, Clouds)	Sentinel-2	**T:** Annual data from 2017 to 2023 **S:** 10m resolution for download	Globally 2017-2023

APPENDIX 4: GEE TOOL AND APP JAVASCRIPT CODE DETAILS

```
var image = ee.Image("projects/ee-yoshimomotancobu2000/assets/ga_ls_wo_fq_myear_3_1987--P37Y_final_frequency/tile0"),
    image2 = ee.Image("projects/ee-yoshimomotancobu2000/assets/ga_ls_wo_fq_myear_3_1987--P37Y_final_frequency/tile1"),
    image3 = ee.Image("projects/ee-yoshimomotancobu2000/assets/ga_ls_wo_fq_myear_3_1987--P37Y_final_frequency/tile10"),
    image4 = ee.Image("projects/ee-yoshimomotancobu2000/assets/ga_ls_wo_fq_myear_3_1987--P37Y_final_frequency/tile11"),
    image5 = ee.Image("projects/ee-yoshimomotancobu2000/assets/ga_ls_wo_fq_myear_3_1987--P37Y_final_frequency/tile12"),
    image6 = ee.Image("projects/ee-yoshimomotancobu2000/assets/ga_ls_wo_fq_myear_3_1987--P37Y_final_frequency/tile13"),
    image7 = ee.Image("projects/ee-yoshimomotancobu2000/assets/ga_ls_wo_fq_myear_3_1987--P37Y_final_frequency/tile14"),
    image8 = ee.Image("projects/ee-yoshimomotancobu2000/assets/ga_ls_wo_fq_myear_3_1987--P37Y_final_frequency/tile15"),
    image9 = ee.Image("projects/ee-yoshimomotancobu2000/assets/ga_ls_wo_fq_myear_3_1987--P37Y_final_frequency/tile2"),
    image10 = ee.Image("projects/ee-yoshimomotancobu2000/assets/ga_ls_wo_fq_myear_3_1987--P37Y_final_frequency/tile3"),
    image11 = ee.Image("projects/ee-yoshimomotancobu2000/assets/ga_ls_wo_fq_myear_3_1987--P37Y_final_frequency/tile4"),
    image12 = ee.Image("projects/ee-yoshimomotancobu2000/assets/ga_ls_wo_fq_myear_3_1987--P37Y_final_frequency/tile5"),
    image13 = ee.Image("projects/ee-yoshimomotancobu2000/assets/ga_ls_wo_fq_myear_3_1987--P37Y_final_frequency/tile6"),
    image14 = ee.Image("projects/ee-yoshimomotancobu2000/assets/ga_ls_wo_fq_myear_3_1987--P37Y_final_frequency/tile7"),
    image15 = ee.Image("projects/ee-yoshimomotancobu2000/assets/ga_ls_wo_fq_myear_3_1987--P37Y_final_frequency/tile8"),
    image16 = ee.Image("projects/ee-yoshimomotancobu2000/assets/ga_ls_wo_fq_myear_3_1987--P37Y_final_frequency/tile9");
//------------------------------------------------------------------------------------------------------------------------------------------------------
// MY 1/Oct/2024 - Initial creation
// MY 18/Mar/2025 - combine with version 3



/*
 * Initialise the app
 */

ui.root.clear();

// Set the root panel's absolute layout, which positions widgets according to positions in the panel (unlik the flow layout).
// should be used with position parameter in the panel style setting and ui.root.add(panel)
ui.root.setLayout(ui.Panel.Layout.absolute());



/*
 * Map layer configuration
 */
```

```
// Create the main map.
var mapPanel = ui.Map({style: {width: '75%', height: '100%', position: 'middle-right'}});
ui.root.add(mapPanel);

// Initialise the map with a specific point (default region).
var initialLonLat = [144.273, -20.842]  // Hughenden //TNT-Ben: Change this to the relevant location
var initialPoint = ee.Geometry.Point(initialLonLat[0], initialLonLat[1]);
mapPanel.centerObject(initialPoint, 15);  // number: the zoom level from 0 to 24
// Initialise basemap type.
mapPanel.setOptions('SATELLITE');



/*
 * Panel set up
 */

// Add a title to a side panel.
// list of label style properties ----> https://developers.google.com/earth-engine/apidocs/ui-label-style
var title = ui.Label('Water Mapping', {fontWeight: 'bold', fontSize: '36px', color: 'blue'});
var toolPanel = ui.Panel([title], 'flow', {width: '20%', position: 'top-left'});
ui.root.add(toolPanel);



// Add Drawing Tools into the mapPanel, so that users can select their area of interest (Draw a shape/rectangle).
var drawingTools = mapPanel.drawingTools();

// Hide the tools so you can add your own.
drawingTools.setShown(false);

// Clear all existing geometries to handle a single geometry.
while (drawingTools.layers().length() > 0) {
  var layer = drawingTools.layers().get(0);
  drawingTools.layers().remove(layer);
}

var dummyGeometry =
```

```
    ui.Map.GeometryLayer({geometries: null, name: 'geometry', color: '#FF0000'});
drawingTools.layers().add(dummyGeometry);

// Initialise a variable to store the drawn geometry
var drawnGeometry;
print(drawnGeometry);

// When drawing is complete, retrieve the rectangle/polygon as an ee.Geometry
drawingTools.onDraw(function() {
  clearErrorMessage();
  drawnGeometry = drawingTools.layers().get(0).getEeObject();  // ?Not sure what .get(0) bit does?
  print("New geometry has been added.");  // Output for verification
  print("The geometry layer's ee.Object is now:");
  print(drawnGeometry);

});

// Create a 'draw area' label and buttons.
var fontsize = '16px';
var fontweight = 'bold';
var text_area = ui.Label('1. Draw Area.', {fontWeight: fontweight, fontSize: fontsize});
toolPanel.add(text_area);

var symbol = {
  rectangle: '🟥',
  polygon: '🔺',
};
var buttonPanel = ui.Panel({
  widgets: [
    ui.Button({
      label: symbol.rectangle + ' Rectangle',
      onClick: drawRectangle
    }),
    ui.Button({
      label: symbol.polygon + ' Polygon',
      onClick: drawPolygon
    })
  ],
```

```
  layout: ui.Panel.Layout.flow('horizontal')  // arrange the buttons side by side
});
toolPanel.add(buttonPanel);



var text_date = ui.Label('2. Set Date.', {fontWeight: fontweight, fontSize: fontsize});
toolPanel.add(text_date);
// Initialise time frame.
// TNT-Ben: what will be the relevant start and end dates and the time step/interval (how many days?)?
var START = ee.Date('2015-06-28');
var NOW = ee.Date(Date.now());
var END = ee.Date(NOW).format();  // convert a date to string.
var interval = 1;
var wm_start = START;

// Add dateslider and calendar.
// ui.DateSlider
// URL: https://developers.google.com/earth-engine/apidocs/ui-dateslider
// default start -> 1 week ago, end -> today, value -> yesterday, period -> one
var errorLabels = []; // Store error message labels
var error_drawArea = 'Please draw your area first (1. Draw Area.)';
var dateSlider = ui.DateSlider({
  start: START,
  value: null,
  period: interval,
  onChange: function(dateRange){  // wm_start updates to the date selected by users
  // Stop execution of the checkbox function if drawnGeometry is undefined.
    if (!drawnGeometry) {
      showErrorMessage(error_drawArea);
      // Undo the change of date to prevent confusion.
       dateSlider.setValue(START, false);
      return;  // Stop execution
    }
    wm_start = ee.Date(dateRange.start());
    print('Start date of the dateslider now:');
    print(wm_start);
  },
  style: {width: '40%'}
```

```
});
toolPanel.add(dateSlider);



//  Add layers.
var text_product = ui.Label('3. Select Products.', {fontWeight: fontweight, fontSize: fontsize});
toolPanel.add(text_product);


// Initialise the error label.
var errorLabel;



// Initialise shared variables (e.g., period).
var period = 7;  // TNT-Ben: this will become 1 later once we have enough satellite inputs



// Label/Layer names
var satellite_image = 'Satellite Image';
var water_map = 'Water Map';

// Dictionary to track active water summary panels
var waterSummaryPanels = {};

// Dynamic World V1: created from Sentinel-2 imagery, 10m-resolution, NRT
// var dwv1_labelname = 'Dynamic World V1 (10m) - collection period of ' + period + ' days';
var dwv1_labelname = 'Sentinel-2 (DWV1)';
var dwv1_link = ui.Label('About Sentinel-2 (DWV1)', {fontSize: '12px', margin: '0 0 0 26px'}, 'https://developers.google.com/earth-engine/datasets/catalog/GOOGLE_DYNAMICWORLD_V1');
var dwv1_si_layername = satellite_image + ' (' + dwv1_labelname + ')';  // satellite image
var dwv1_wm_layername = water_map + ' (' + dwv1_labelname + ')';  // water map
var dwv1_res = 10;
var sent2_error = null;
// Add Checkbox to toggle the Dynamic World map layer.
var dwCheckbox = ui.Checkbox({
label: dwv1_labelname,
value: false,
onChange: function(checked) {
 // Remove any existing error message before processing.
```

```
clearErrorMessage();

if (checked) {
 // Stop execution of the checkbox function if drawnGeometry is undefined.
 if (!drawnGeometry) {
  showErrorMessage(error_drawArea);
  // Uncheck the checkbox to prevent confusion.
  dwCheckbox.setValue(false, false);
  return;  // Stop execution
 }

 // Deselect the geometry layer.
 drawingTools.layers().get(0).setShown(0);

 //TNT: Undisplay all process messages.

 print('Extracting the DWV1 layer of start date:');
 print(wm_start);
 print('with the period of ' + period + ' days.');
 var linkedImg = dwv1_s2(wm_start, period, drawnGeometry);  // this 'wm_start' keeps updating as you change the date on the slider
 // Check if linkedImg exists.
 if (linkedImg == null) {
  // Display error message in the toolPanel.
  var sent2_error = 'Sentinel-2 (DWV1): No data available for the selected area and/or date.';
  showErrorMessage(sent2_error);
  dwCheckbox.setValue(false, false);
  return;
 }

 var dwRgbHillshade = dwv1(linkedImg);
 // mapPanel.addLayer(linkedImg, {min: 0, max: 3000, bands: ['B4', 'B3', 'B2']}, 'Sentinel-2 L1C');
 var si_layer = mapPanel.addLayer(linkedImg, {min: 0, max: 3000, bands: ['B4', 'B3', 'B2']}, dwv1_si_layername);
 si_layer.setShown(0);
 mapPanel.addLayer(dwRgbHillshade, {min: 0, max: 0.65}, dwv1_wm_layername);
 print(drawingTools.layers());

 // Extract the number of pixels for each class.
 var water_pixel_count = dwv1_pix_count_water(wm_start, period, drawnGeometry);
```

```
    // Calculate the total area of water body in m^2.
    var one_pixel_area = dwv1_res * dwv1_res;
    var total_water_area = ee.Number(water_pixel_count).multiply(one_pixel_area);
    print('The total area of water in your drawn polygon is...');
    print(total_water_area);
    print('m²');
    // Evaluate the total water area to get the actual value
    total_water_area.evaluate(function(result) {
      // Convert the evaluated value to a string
      var total_water_area_str = result.toString();

      // Display the information in the UI
      print(dwv1_labelname + ': ' + total_water_area_str + ' m²');

      // Create and add the label with the evaluated string
      var text_ws_s2 = ui.Label(dwv1_labelname + ': ' + total_water_area_str + ' m²', {fontSize: '14px', color: '#419BDF'});
      // Store in dictionary
      waterSummaryPanels[dwv1_labelname] = text_ws_s2;
      waterSummaryPanel.add(text_ws_s2);
    });

  } else {
    // Remove only this product's summary
    if (waterSummaryPanels[dwv1_labelname]) {
      waterSummaryPanel.remove(waterSummaryPanels[dwv1_labelname]);
      delete waterSummaryPanels[dwv1_labelname]; // Remove reference
    }
    removeLayer(dwv1_si_layername);
    removeLayer(dwv1_wm_layername);  // remove layers with the specified names
  }
}
});
toolPanel.add(dwCheckbox);
toolPanel.add(dwv1_link);



// Sentinel1: 10m-resolution
// var senti1_labelname = 'Sentinel1 (??m) - collection period of ' + period + ' days';
```

```
var senti1_labelname = 'Sentinel-1';
var senti1_res = 10;
var senti1_si_layername = satellite_image + ' (' + senti1_labelname + ')';  // satellite image
var senti1_wm_layername = water_map + ' (' + senti1_labelname + ')';  // water map

// Add Checkbox to toggle the Sentinel-1 map layer.
var senti1Checkbox = ui.Checkbox({
label: senti1_labelname,
value: false,
onChange: function(checked) {
 // Remove any existing error message before processing.
 clearErrorMessage();

 if (checked) {
  // Stop execution of the checkbox function if drawnGeometry is undefined.
  if (!drawnGeometry) {
   showErrorMessage(error_drawArea);
   // Uncheck the checkbox to prevent confusion.
   senti1Checkbox.setValue(false, false);
   return;  // Stop execution
  }
  // Deselect the geometry layer.
  drawingTools.layers().get(0).setShown(0);

  //TNT: Undisplay all process messages.

  print('Extracting the Sentinel1 layer of start date:');
  print(wm_start);
  print('with the period of ' + period + ' days.');
  // Set "drawnGeometry" as an input variable to display the layer within this polygon.
  var linkedImg = senti1Img(wm_start, period, drawnGeometry);  // this 'wm_start' keeps updating as you change the date on the slider
  // Check if linkedImg exists.
  if (linkedImg == null) {
   senti1Checkbox.setValue(false, false);
   return;
  }
  var senti1 = senti1WaterMask(linkedImg);
  // var senti1_si_layername = satellite_image + ' (' + senti1_labelname + ')';  // satellite image
```

```
  // var senti1_wm_layername = water_map + ' (' + senti1_labelname + ')';  // water map
  var si_layer = mapPanel.addLayer(linkedImg, {min:-25,max:0}, senti1_si_layername);
  si_layer.setShown(0);
  mapPanel.addLayer(senti1, {palette:['Blue']}, senti1_wm_layername);
  print(drawingTools.layers());
  // Extract the number of pixels for each class.
  var water_pixel_count = senti1_pix_count_water(linkedImg);
  // Calculate the total area of water body in m^2.
  var one_pixel_area = senti1_res * senti1_res;
  var total_water_area = ee.Number(water_pixel_count).multiply(one_pixel_area);
  print('The total area of water in your drawn polygon is...');
  print(total_water_area);
  print('m²');
  // Evaluate the total water area to get the actual value
  total_water_area.evaluate(function(result) {
     // Convert the evaluated value to a string
     var total_water_area_str = result.toString();


     // Display the information in the UI
     print(senti1_labelname + ': ' + total_water_area_str + ' m²');


     // Create and add the label with the evaluated string
     var text_ws_s1 = ui.Label(senti1_labelname + ': ' + total_water_area_str + ' m²', {fontSize: '14px', color: 'blue'});
     // Store in dictionary
     waterSummaryPanels[senti1_labelname] = text_ws_s1;
     waterSummaryPanel.add(text_ws_s1);
  });


 } else {
  // Remove only this product's summary
  if (waterSummaryPanels[senti1_labelname]) {
    waterSummaryPanel.remove(waterSummaryPanels[senti1_labelname]);
    delete waterSummaryPanels[senti1_labelname]; // Remove reference
  }
  print(satellite_image, senti1_labelname);
  print(senti1_si_layername);
  removeLayer(senti1_si_layername);
  removeLayer(senti1_wm_layername);  // remove layers with the specified names
```

```
  }
}
});
toolPanel.add(senti1Checkbox);



// Landsat - WOfS (DEA Water Observations Multi-Year): 30m-resollution
var landsat_labelname = 'WOfS (1986-2024)';
var landsat_link = ui.Label('About WOfS (1986-2024)', {fontSize: '12px', margin: '0 0 0 26px'}, 'https://knowledge.dea.ga.gov.au/data/version-history/dea-water-observations-landsat-2.1.5/');
var landsat_wm_layername = water_map + ' (' + landsat_labelname + ')';  // water map
var legendPanel = null;
// Add Checkbox to toggle the Landsat map layer.
var landsatCheckbox = ui.Checkbox({
label: landsat_labelname,
value: false,
onChange: function(checked) {
 // Remove any existing error message before processing.
 clearErrorMessage();

 if (checked) {
  // Stop execution of the checkbox function if drawnGeometry is undefined.
  if (!drawnGeometry) {
   showErrorMessage(error_drawArea);
   // Uncheck the checkbox to prevent confusion.
   landsatCheckbox.setValue(false, false);
   return;  // Stop execution
  }
  // Deselect the geometry layer.
  drawingTools.layers().get(0).setShown(0);



  print('Displaying Water Summary layer of DEA Water Observations Multi-Year 1986-2024 (WOfS).');


  // Merge all 16 entries into one raster layer and Clip.
  var WOfS_Img = WOfS().clip(drawnGeometry);
  // Set "drawnGeometry" as an input variable to display the layer within this polygon.
```

```
// Mask out values equal to 0
var masked_image = WOfS_Img.updateMask(WOfS_Img.gt(0));

// linearly interpolated between the maximum and minimum values in the visualization parameters
// (or defaults according to the band type, as described previously). For example, pixels less than
// or equal to the minimum will be displayed with the first color in the list; pixels greater than or
// equal to the maximum will be displayed with the last color in the list. Intermediate colors are
// linearly stretched to intermediate pixel values.
//TNT: more gradients i.e., colours?; create a colour palette like the one in DEA Map
var gradientColors = ['#FFFFFF',  // 0%
                '#a6e300',  // 12.5%
                '#00e32d',  // 25%
                '#00e384',  // 37.5%
                '#00E3C8',  // 50%
                '#00d1d7',  // 62.5%
                '#0097e3',  // 75%
                '#001fe3',  // 87.5%
                '#5700E3',  // 100%
                ];

// Visualization parameters
var visParams = {
  min: 0,       // Minimum value to start the color gradient
  max: 1,       // Maximum value
  palette: gradientColors
};

// Add the masked image layer to the map
mapPanel.addLayer(masked_image, visParams, landsat_wm_layername);
print(drawingTools.layers());
// Add color bar of water summary 0-100% -> refer to WOfS_myear
// Define the color bar for the WOfS product
var colorBar = ui.Thumbnail({
  image: ee.Image.pixelLonLat().select(0), // Create a linear gradient for the bar
  params: {
    bbox: [0, 0, 1, 0.05],              // Aspect ratio and size of the color bar
    dimensions: '200x20',               // Dimensions of the color bar
    min: 0,                         // Minimum value of the gradient
```

```
    max: 1,                         // Maximum value of the gradient
    palette: gradientColors         // Use the WOfS gradient colors
  },
  style: {
    stretch: 'horizontal',          // Stretch the color bar horizontally
    margin: '10px 0 0 0'            // Add some margin above the bar
  }
});

// Define labels for the color bar
var colorBarLabels = ui.Panel({
  widgets: [
    ui.Label('<0%', {margin: '0px', fontSize: '10px'}),  // Start label; values equal to 0 have already been excluded
    ui.Label('50%', {margin: '0px', textAlign: 'center', stretch: 'horizontal', fontSize: '10px'}), // Mid label
    ui.Label('100%', {margin: '0px', fontSize: '10px'}) // End label
  ],
  layout: ui.Panel.Layout.flow('horizontal'),       // Arrange labels horizontally
  style: {stretch: 'horizontal'}                    // Stretch the labels
});

// Assign to global `legendPanel` so we can remove it later
legendPanel = ui.Panel({
  widgets: [
    ui.Label(landsat_labelname + ': ', {fontSize: fontsize}), // Legend title
    // what percentage of clear observations were detected as wet (i.e. the ratio of wet to clear as a percentage) between 1986 and 2024
    ui.Label('Water Frequency', {margin: '0px', textAlign: 'center', stretch: 'horizontal', fontSize: '12px'}),
    colorBar,                                       // Add the color bar
    colorBarLabels                                  // Add the labels
  ],
  style: {
    position: 'bottom-left',
    padding: '8px',
    margin: '0px 0px 0px 0px',
    backgroundColor: 'white'
  }
});

// Store in dictionary
```

```
    waterSummaryPanels[landsat_labelname] = legendPanel;
    waterSummaryPanel.add(legendPanel);

  } else {
    // Remove only this product's summary
    if (waterSummaryPanels[landsat_labelname]) {
      waterSummaryPanel.remove(waterSummaryPanels[landsat_labelname]);
      delete waterSummaryPanels[landsat_labelname]; // Remove reference
    }
    print(landsat_wm_layername);
    removeLayer(landsat_wm_layername);

  }
}
});
toolPanel.add(landsatCheckbox);
toolPanel.add(landsat_link);

// Create a panel for the Water Summary section
var text_ws = ui.Label('4. Water Summary', {fontWeight: fontweight, fontSize: fontsize});
toolPanel.add(text_ws);
var waterSummaryPanel = ui.Panel([]);
toolPanel.add(waterSummaryPanel);



// Clear GeometryLayers when clicked - Rest button.
var geometryResetButton = ui.Button({
  label: 'Reset',
  onClick: resetAll,
});
toolPanel.add(geometryResetButton);




// Replace the root with a SplitPanel that contains the tool and map.
ui.root.clear();
ui.root.add(ui.SplitPanel(toolPanel, mapPanel)); //----> this code will be useful when you want to move the split between map and tool panel with your cursor
```

```
//##########################################################################################################################################
####################

// Library of functions

//##########################################################################################################################################
####################


//############ Dynamic World V1 ############

// Return Sentinel-2 L1C: Level-1C, radiometric and geometric Top-of-Atmosphere reflectance product

function dwv1_s2(start, period, geometry){

 var end = start.advance(period, 'day');

 print(start, end);

 var colFilter = ee.Filter.and(

    ee.Filter.bounds(geometry),  // this also changes depending on users preference

    ee.Filter.date(start, end));

 var dwCol = ee.ImageCollection('GOOGLE/DYNAMICWORLD/V1').filter(colFilter);  // DW NRT collection


 // Check if the 'dwCol' is empty.

 if (dwCol.size().eq(0).getInfo()) {

  // Stop further execution.

  linkedImg = null;

  return linkedImg;

 }


 var s2Col = ee.ImageCollection('COPERNICUS/S2_HARMONIZED');

 // Link DW and S2 source images.

 var linkedCol = dwCol.linkCollection(s2Col, s2Col.first().bandNames());

 // Get example DW image with linked S2 image.

 var linkedImg = ee.Image(linkedCol.first()).clip(geometry);

 print("Debug: ", linkedImg);

 return linkedImg;

}
```

```
// Return Dynamic World V1 - label hillshade.
function dwv1(linkedImg){
  // Create a visualization that blends DW class label with probability.
  // Define list pairs of DW LULC label and color.
  var CLASS_NAMES = [
      'water', 'trees', 'grass', 'flooded_vegetation', 'crops',
      'shrub_and_scrub', 'built', 'bare', 'snow_and_ice'];  // we want to show only 'water'
  // var CLASS_NAMES = [
  //     'water']; -> this does not work

  var VIS_PALETTE = [
      '419bdf', '397d49', '88b053', '7a87c6', 'e49635', 'dfc35a', 'c4281b',
      'a59b8f', 'b39fe1'];
  // var VIS_PALETTE = [
  //     '419bdf']; -> this does not work

  // Create an RGB image of the label (most likely class) on [0, 1].
  var dwRgb = linkedImg
      .select('label')
      .visualize({min: 0, max: 8, palette: VIS_PALETTE})
      .divide(255);

  // Get the most likely class probability.
  var top1Prob = linkedImg.select(CLASS_NAMES).reduce(ee.Reducer.max());

  // Create a hillshade of the most likely class probability on [0, 1];
  var top1ProbHillshade =
      ee.Terrain.hillshade(top1Prob.multiply(100))
      .divide(255);

  // Combine the RGB image with the hillshade.
  var dwRgbHillshade = dwRgb.multiply(top1ProbHillshade);
  return dwRgbHillshade;
}

// Return pixel count for water class.
function dwv1_pix_count_water(start, period, geometry){
  var end = start.advance(period, 'day');
```

```
print(start, end);
var colFilter = ee.Filter.and(
   ee.Filter.bounds(geometry),  // this also changes depending on users preference
   ee.Filter.date(start, end));
var dwCol = ee.ImageCollection('GOOGLE/DYNAMICWORLD/V1').filter(colFilter);
print("Debug1: ", dwCol);
// Create a mode composite.
var classification = dwCol.select('label');
var dwComposite = classification.reduce(ee.Reducer.mode());
var dwComposite = dwComposite.rename(['classification']);
var pixelCountStats = dwComposite.reduceRegion({
  reducer: ee.Reducer.frequencyHistogram().unweighted(),  // compute the counts for all unique pixel values in the image
  // .unweighted(): pixels are included if their centroid is in the region and the image's mask is non-zero.
  geometry: geometry,
  scale: 10,  // pixel resolution in meters
  maxPixels: 1e10
  });

var pixelCounts = ee.Dictionary(pixelCountStats.get('classification'));
print(pixelCounts);
// Check if '0' key exists before accessing it.
var waterCount = ee.Algorithms.If(pixelCounts.contains('0'), pixelCounts.get('0'), 0);
print(waterCount);
return waterCount

// // Format the results to make it more readable.
// var classLabels = ee.List([
//     'water', 'trees', 'grass', 'flooded_vegetation', 'crops',
//     'shrub_and_scrub', 'built', 'bare', 'snow_and_ice'
//     ]);

// // Rename keys with class names.
// var pixelCountsFormatted = pixelCounts.rename(
//   pixelCounts.keys(), classLabels);
// print(pixelCountsFormatted);
// return pixelCountsFormatted;
}
```

```
//############ Sentinel1 (the code developed by Ben) ############

// Return Sentinel1 Image.

function senti1Img(start, period, geometry){

  var end = start.advance(period, 'day');

  print(start, end);

  var colFilter = ee.Filter.and(

     ee.Filter.bounds(geometry),

     ee.Filter.date(start, end));

  var collection = ee.ImageCollection('COPERNICUS/S1_GRD')  // URL: https://developers.google.com/earth-
engine/datasets/catalog/COPERNICUS_S1_GRD

       .filter(ee.Filter.listContains('transmitterReceiverPolarisation', 'VH'))

       .filter(ee.Filter.eq('instrumentMode', 'IW'))

       .filter(ee.Filter.or(ee.Filter.eq('orbitProperties_pass', 'DESCENDING'),ee.Filter.eq('orbitProperties_pass', 'ASCENDING')))

       .filter(colFilter);


  // Check if the 'dwCol' is empty.

  if (collection.size().eq(0).getInfo()) {

    // Display error message in the toolPanel.

    var sent1_error = 'Sentinel-1: No data available for the selected area and/or date.';

    showErrorMessage(sent1_error);

    // Stop further execution.

    linkedImg = null;

    return linkedImg;

  }


  var linkedImg = collection.select('VH').mosaic().clip(geometry);

  return linkedImg;

}


// Return Sentinel1 Water Mask.

function senti1WaterMask(linkedImg){

  var filtered_VH = ee.Image(toDB(RefinedLee(toNatural(linkedImg))));

  var water_VH = filtered_VH.lt(-19);

  var water_mask_VH = water_VH.updateMask(water_VH.eq(1));

  return water_mask_VH;

}
```

```
// Return pixel count for water class.
function senti1_pix_count_water(linkedImg){
  var filtered_VH = ee.Image(toDB(RefinedLee(toNatural(linkedImg))));
  var water_VH = filtered_VH.lt(-19);
  var water_mask_VH = water_VH.updateMask(water_VH.eq(1));

  var geometry = linkedImg.geometry();
  // Count the number of "1" pixels using reduceRegion.
  var waterCount = water_mask_VH.reduceRegion({
    reducer: ee.Reducer.sum(),
    geometry: geometry,
    scale: 10, // pixel resolution in meters
    maxPixels: 1e10
  }).get('sum');

  print(waterCount);
  return waterCount
}


// Convert the input image?? to dB.
function toDB(img) {
  return ee.Image(img).log10().multiply(10.0);
}


// Convert the input image?? from d?
function toNatural(img) {
  return ee.Image(10.0).pow(img.select(0).divide(10.0));
}

// Aplly a Refined Lee Speckle filter as coded in the SNAP 3.0 S1TBX.
//https://github.com/senbox-org/s1tbx/blob/master/s1tbx-op-sar-
processing/src/main/java/org/esa/s1tbx/sar/gpf/filtering/SpeckleFilters/RefinedLee.java
//Adapted by Guido Lemoine
// by Guido Lemoine
function RefinedLee(img) {
  // img must be in natural units, i.e. not in dB!
```

```
// Set up 3x3 kernels
var weights3 = ee.List.repeat(ee.List.repeat(1,3),3);
var kernel3 = ee.Kernel.fixed(3,3, weights3, 1, 1, false);

var mean3 = img.reduceNeighborhood(ee.Reducer.mean(), kernel3);
var variance3 = img.reduceNeighborhood(ee.Reducer.variance(), kernel3);

// Use a sample of the 3x3 windows inside a 7x7 windows to determine gradients and directions
var sample_weights = ee.List([[0,0,0,0,0,0,0], [0,1,0,1,0,1,0],[0,0,0,0,0,0,0], [0,1,0,1,0,1,0], [0,0,0,0,0,0,0], [0,1,0,1,0,1,0],[0,0,0,0,0,0,0]]);

var sample_kernel = ee.Kernel.fixed(7,7, sample_weights, 3,3, false);

// Calculate mean and variance for the sampled windows and store as 9 bands
var sample_mean = mean3.neighborhoodToBands(sample_kernel);
var sample_var = variance3.neighborhoodToBands(sample_kernel);

// Determine the 4 gradients for the sampled windows
var gradients = sample_mean.select(1).subtract(sample_mean.select(7)).abs();
gradients = gradients.addBands(sample_mean.select(6).subtract(sample_mean.select(2)).abs());
gradients = gradients.addBands(sample_mean.select(3).subtract(sample_mean.select(5)).abs());
gradients = gradients.addBands(sample_mean.select(0).subtract(sample_mean.select(8)).abs());

// And find the maximum gradient amongst gradient bands
var max_gradient = gradients.reduce(ee.Reducer.max());

// Create a mask for band pixels that are the maximum gradient
var gradmask = gradients.eq(max_gradient);

// duplicate gradmask bands: each gradient represents 2 directions
gradmask = gradmask.addBands(gradmask);

// Determine the 8 directions
var directions = sample_mean.select(1).subtract(sample_mean.select(4)).gt(sample_mean.select(4).subtract(sample_mean.select(7))).multiply(1);
directions =
directions.addBands(sample_mean.select(6).subtract(sample_mean.select(4)).gt(sample_mean.select(4).subtract(sample_mean.select(2))).multiply(
2));
directions =
directions.addBands(sample_mean.select(3).subtract(sample_mean.select(4)).gt(sample_mean.select(4).subtract(sample_mean.select(5))).multiply(
3));
```

```
 directions =
directions.addBands(sample_mean.select(0).subtract(sample_mean.select(4)).gt(sample_mean.select(4).subtract(sample_mean.select(8))).multiply(
4));

 // The next 4 are the not() of the previous 4

 directions = directions.addBands(directions.select(0).not().multiply(5));

 directions = directions.addBands(directions.select(1).not().multiply(6));

 directions = directions.addBands(directions.select(2).not().multiply(7));

 directions = directions.addBands(directions.select(3).not().multiply(8));


 // Mask all values that are not 1-8

 directions = directions.updateMask(gradmask);


 // "collapse" the stack into a singe band image (due to masking, each pixel has just one value (1-8) in it's directional band, and is otherwise masked)

 directions = directions.reduce(ee.Reducer.sum());


 //var pal = ['ffffff','ff0000','ffff00', '00ff00', '00ffff', '0000ff', 'ff00ff', '000000'];

 //Map.addLayer(directions.reduce(ee.Reducer.sum()), {min:1, max:8, palette: pal}, 'Directions', false);


 var sample_stats = sample_var.divide(sample_mean.multiply(sample_mean));


 // Calculate localNoiseVariance

 var sigmaV = sample_stats.toArray().arraySort().arraySlice(0,0,5).arrayReduce(ee.Reducer.mean(), [0]);


 // Set up the 7*7 kernels for directional statistics

 var rect_weights = ee.List.repeat(ee.List.repeat(0,7),3).cat(ee.List.repeat(ee.List.repeat(1,7),4));


 var diag_weights = ee.List([[1,0,0,0,0,0,0], [1,1,0,0,0,0,0], [1,1,1,0,0,0,0],

   [1,1,1,1,0,0,0], [1,1,1,1,1,0,0], [1,1,1,1,1,1,0], [1,1,1,1,1,1,1]]);


 var rect_kernel = ee.Kernel.fixed(7,7, rect_weights, 3, 3, false);

 var diag_kernel = ee.Kernel.fixed(7,7, diag_weights, 3, 3, false);


 // Create stacks for mean and variance using the original kernels. Mask with relevant direction.

 var dir_mean = img.reduceNeighborhood(ee.Reducer.mean(), rect_kernel).updateMask(directions.eq(1));

 var dir_var = img.reduceNeighborhood(ee.Reducer.variance(), rect_kernel).updateMask(directions.eq(1));


 dir_mean = dir_mean.addBands(img.reduceNeighborhood(ee.Reducer.mean(), diag_kernel).updateMask(directions.eq(2)));

 dir_var = dir_var.addBands(img.reduceNeighborhood(ee.Reducer.variance(), diag_kernel).updateMask(directions.eq(2)));
```

```
  // and add the bands for rotated kernels
  for (var i=1; i<4; i++) {
    dir_mean = dir_mean.addBands(img.reduceNeighborhood(ee.Reducer.mean(), rect_kernel.rotate(i)).updateMask(directions.eq(2*i+1)));
    dir_var = dir_var.addBands(img.reduceNeighborhood(ee.Reducer.variance(), rect_kernel.rotate(i)).updateMask(directions.eq(2*i+1)));
    dir_mean = dir_mean.addBands(img.reduceNeighborhood(ee.Reducer.mean(), diag_kernel.rotate(i)).updateMask(directions.eq(2*i+2)));
    dir_var = dir_var.addBands(img.reduceNeighborhood(ee.Reducer.variance(), diag_kernel.rotate(i)).updateMask(directions.eq(2*i+2)));
  }

  // "collapse" the stack into a single band image (due to masking, each pixel has just one value in it's directional band, and is otherwise masked)
  dir_mean = dir_mean.reduce(ee.Reducer.sum());
  dir_var = dir_var.reduce(ee.Reducer.sum());

  // A finally generate the filtered value
  var varX = dir_var.subtract(dir_mean.multiply(dir_mean).multiply(sigmaV)).divide(sigmaV.add(1.0));

  var b = varX.divide(dir_var);

  var result = dir_mean.add(b.multiply(img.subtract(dir_mean)));
  return(result.arrayFlatten([['sum']]));
}

function WOfS(){
  // Add images to an ImageCollection
  var imageCollection = ee.ImageCollection([image, image2, image3, image4, image5, image6, image7,
                                image8, image9, image10, image11, image12, image13, image14, image15, image16]);

  // Merge images using .mosaic()
  var mergedImage = imageCollection.mosaic();
  return mergedImage;
}


//############ Layout ############
// Clear map layers of specified names.
function removeLayer(layername) {
  print(layername);
  var layers = mapPanel.layers();
```

```
  var names = [];
  layers.forEach(function(layer) {
    var lay_name = layer.getName();
    names.push(lay_name);
  })

  var index = names.indexOf(layername);
  if (index > -1) {
    var layer = layers.get(index);
    mapPanel.remove(layer);
    print('"'+layername+'"'+' has been removed.');
  } else {
    print('Layer '+layername+' not found')
  }
}




// Clear all layers on the map.
function clearAllMapLayers() {
  var mapLayers = mapPanel.layers();
  // print(mapLayers);
  // print(mapLayers.length())
  if (mapLayers.length() > 0) {
    mapLayers.reset();
  print(mapLayers);
  }
}

// Clear all input geometries and layers on the map.
function resetAll() {
  clearErrorMessage();
  var layers = drawingTools.layers();
  //layers.get(0).geometries().remove(layers.get(0).geometries().get(0));  // this only remove the first geometry not all of them
  var geometries = layers.get(0).geometries();
  // Loop through each geometry in the first layer and remove it.
  while (geometries.length() > 0) {
    geometries.remove(geometries.get(0));
```

```
    print("Removing the geometry...");
    print(geometries.get(0))

    drawnGeometry = drawingTools.layers().get(0).getEeObject();
    print("Your geometry/ries has/have been removed.");
    print(drawnGeometry);
  }
  clearAllMapLayers();
  // Uncheck all products & set date to initial
  dwCheckbox.setValue(false);
  senti1Checkbox.setValue(false);
  landsatCheckbox.setValue(false);
  // Initialise the water summary panel.
  waterSummaryPanel.clear();

  // Stop and deactivate drawing mode.
  drawingTools.setShape(null);

  // Switch cursor to hand (grabbing mode).
  mapPanel.style().set('cursor', 'hand');

  print('Tool reset completed.');

}

// Show an error message on the toolPanel.
function showErrorMessage(msg) {
  var errorLabel = ui.Label(msg, {color: 'red', fontWeight: 'bold'});
  toolPanel.add(errorLabel);
  errorLabels.push(errorLabel); // Track error messages
}

// Clear the existing error message
function clearErrorMessage() {
  while (errorLabels.length > 0) {
    var label = errorLabels.pop();
    toolPanel.remove(label);
  }
```

```
}

function clearGeometry() {
  var layers = drawingTools.layers();
  layers.get(0).geometries().remove(layers.get(0).geometries().get(0));
}

function drawRectangle() {
  clearGeometry();
  clearErrorMessage();
  drawingTools.setShape('rectangle');
  drawingTools.draw();
}

function drawPolygon() {
  clearGeometry();
  clearErrorMessage();
  drawingTools.setShape('polygon');
  drawingTools.draw();
}



// function updateWaterSummary(widget) {
//   waterSummaryPanel.clear();
//   var text_ws = ui.Label('Water Summary', {fontWeight: fontweight, fontSize: fontsize});
//   waterSummaryPanel.add(text_ws);
//   waterSummaryPanel.add(widget);

// } ** this function only shows one analysis (i.e., one product's water summary)

// function drawPolygon() {
//   // Set the shape to 'polygon' and start drawing
//   drawingTools.setShape('polygon');

//   // When drawing is complete, retrieve the polygon as an ee.Geometry
//   drawingTools.onDraw(function() {
//     drawnGeometry = drawingTools.layers().get(0).getEeObject();  // ?Not sure what .get(0) bit does?
//     print("New geometry has been added.");  // Output for verification
```

```
//    print("The geometry layer's ee.Object is now:");
//    print(drawnGeometry);
//    //print(mapPanel.GeometryLayer)
//    //drawnGeometry.setShown(false);  // error
//    //mapPanel.GeometryLayer.setShown(false);  // error
//    // Now `drawnGeometry` is an ee.Geometry.Polygon and can be used in the app

//    //drawingTools.layers().get(0).setOpacity(0); -> error

//    // Add the outline of area drawn on the map by users.
//    // var drawnGeometryFC = ee.FeatureCollection(drawnGeometry);
//    // var empty = ee.Image().byte();
//    // var outline = empty.paint({
//    //   featureCollection: drawnGeometryFC,
//    //   color: 1,
//    //   width: 3
//    // });
//    // mapPanel.addLayer(outline, {palette: 'red'}, 'Outline');



//    // print(drawingTools.layers().get(0))
//    // drawingTools.layers().get(0).setVisParams({
//    //   opacity: 0,
//    //   width: 1,          // Outline width
//    // });


//  });
// }
```