Coastal Change Analysis Using Satellite Imagery and Machine Learning

Open report OR/26/058

BRITISH GEOLOGICAL SURVEY

OPEN REPORT OR/26/058

Keywords

Shoreline Extraction, Remote Sensing, Earth Observation, Machine Learning, Artificial Neural Networks, Sentinel-2, Google Earth Engine, Coastal Geomorphology.

Cover picture: View of the Campi Flegrei coastline via Wikimedia Commons.

Bibliographical reference

E McAllister. 2026. *Coastal Change Analysis Using Satellite Imagery and Machine Learning. British Geological Survey Open Report* OR/26/058. 25pp.

Coastal Change Analysis Using Satellite Imagery and Machine Learning

E McAllister

Keyworth, Nottingham British Geological Survey 2026

BRITISH GEOLOGICAL SURVEY

The full range of our publications is available from the BGS shop at Nottingham and Cardiff (Welsh publications only). Shop online at https://shop.bgs.ac.uk/

The London Information Office also maintains a reference collection of BGS publications, including fossils, for consultation.

We publish an annual catalogue of our maps and other publications; this catalogue is available online or from the BGS shop.

The British Geological Survey carries out the geological survey of Great Britain and Northern Ireland (the latter as an agency service for the government of Northern Ireland), and of the surrounding continental shelf, as well as basic research projects. It also undertakes programmes of technical aid in geology in developing countries.

The British Geological Survey is a component body of UK Research and Innovation.

British Geological Survey offices

Nicker Hill, Keyworth,
Nottingham NG12 5GG
Tel 0115 936 3100

BGS Central Enquiries Desk
Tel 0115 936 3143
email enquiries@bgs.ac.uk

BGS Sales
Tel 0115 936 3241
email sales@bgs.ac.uk

The Lyell Centre, Research Avenue South,
Edinburgh EH14 4AP
Tel 0131 667 1000
email scotsales@bgs.ac.uk

Natural History Museum, Cromwell Road,
London SW7 5BD
Tel 020 7589 4090
Tel 020 7942 5344/45
email bgslondonstaff@bgs.ac.uk

Cardiff University, Main Building, Park Place,
Cardiff CF10 3AT
Tel 029 2167 4280

Geological Survey of Northern Ireland, 7th Floor,
Adelaide House, 39-49 Adelaide Street, Belfast, BT2 8FD
Tel 0289 038 8462
www2.bgs.ac.uk/gsni/

Natural Environment Research Council, Polaris House,
North Star Avenue, Swindon SN2 1EU
Tel 01793 411500   Fax 01793 411501
www.nerc.ac.uk

UK Research and Innovation, Polaris House,
Swindon SN2 1FL
Tel   01793 444000
www.ukri.org

Website: https://www.bgs.ac.uk
Shop online: https://shop.bgs.ac.uk/

Foreword

This report describes the methodology and implementation of a shoreline extraction workflow developed using satellite imagery. The study focuses on deriving shoreline positions from Sentinel-2 data and organising the resulting outputs into a structured and consistent dataset suitable for further analysis and interpretation.

The methods used to extract shoreline features have been developed and applied through a combination of remote sensing techniques and geospatial data processing. The workflow includes data acquisition, preprocessing (including cloud masking and image selection), shoreline detection, and post-processing steps to refine and organise the extracted shoreline outputs.

This document outlines the rationale behind the workflow design, the data sources used, and the key processing steps involved. It also provides an overview of the outputs generated and how they can be utilised for shoreline analysis, monitoring coastal change, and supporting broader geospatial applications.

Acknowledgements

The work was supported through the funding from the Foreign, Commonwealth & Development Office (FCDO) and the Department for Science, Innovation and Technology (DSIT).

Contents

FIGURES

Summary

This report describes the development and application of a shoreline extraction workflow using Sentinel-2 satellite imagery. The study focuses on identifying shoreline positions through remote sensing techniques and processing the resulting data into a structured and usable geospatial format for coastal analysis.

The methodology includes data acquisition from Sentinel-2 imagery, followed by preprocessing steps such as cloud masking, scene selection, and band selection to improve water–land discrimination. Shoreline detection is performed using spectral-based classification methods to separate land and water pixels. Post-processing techniques are then applied to refine the extracted shoreline, including noise reduction, smoothing, and organisation of outputs into consistent spatial datasets.

The workflow is implemented using geospatial tools within a cloud-based processing environment and GIS software to enable efficient and reproducible analysis. The resulting shoreline datasets support the analysis of coastal features and changes over time and demonstrate the potential of Earth observation data for shoreline monitoring and geospatial applications.

1 Introduction

Coastal shorelines are dynamic environments influenced by a range of natural processes, including erosion, sediment transport, tidal fluctuations, and sea-level change. These processes result in continuous spatial and temporal variation in shoreline position, making consistent monitoring essential for understanding coastal behaviour and managing coastal risk (Boak & Turner, 2005).

Traditional shoreline surveying methods, such as field-based GPS surveys or aerial photogrammetry, can provide high accuracy but are often labour-intensive, time-consuming, and limited in spatial and temporal coverage. In contrast, satellite remote sensing offers a scalable and cost-effective alternative for monitoring coastal change over large areas and extended time periods. Multispectral satellite datasets, such as Sentinel-2, enable regular observations of coastal environments with sufficient spatial and temporal resolution for shoreline analysis (McAllister *et al.,* 2022).

In this workflow, Sentinel-2 imagery is used to derive shoreline positions through automated land-water classification. By identifying the boundary between land and water, shoreline positions can be consistently extracted across multiple time steps. These shoreline positions are then used to generate time-series data, enabling the analysis of coastal change patterns, including erosion and accretion trends, across the study area.

2 Objectives

The primary objective of this workflow is to demonstrate a reproducible approach for deriving and analysing shoreline change using satellite imagery and machine learning techniques. The workflow integrates geospatial data processing with supervised classification to support automated shoreline extraction and temporal analysis.

The workflow aims to:

- Extract shoreline positions from Sentinel-2 imagery using automated classification techniques
- Apply an Artificial Neural Network (ANN) to classify landcover based on multispectral spectral signatures
- Generate time-series representations of shoreline movement across multiple acquisition dates
- Quantify spatial patterns of shoreline change using transect-based analysis
- Provide a scalable and repeatable framework for coastal monitoring that can be applied to different geographic regions and time periods

3 Methodology

The methodology describes the end-to-end workflow used to derive shoreline positions from Sentinel-2 imagery and analyse coastal change over time. The process integrates cloud-based data acquisition, preprocessing, machine learning-based classification, and geospatial analysis to extract shoreline features and quantify their temporal variation.

The workflow is designed to be repeatable and scalable, enabling consistent analysis across multiple time periods and geographic locations.

3.1 DATA SOURCES

The primary data source used in this study is Sentinel-2 satellite imagery, provided by the European Space Agency (ESA). Sentinel-2 offers multispectral optical imagery with a spatial resolution of up to 10 metres for visible and near-infrared bands, making it suitable for coastal and shoreline analysis (Drusch *et al.,* 2012).

Imagery was accessed through Google Earth Engine, which provides cloud-based access to large geospatial datasets and enables efficient filtering, preprocessing, and sampling of satellite data environment (Gorelick et al., 2017). Images were selected based on spatial coverage of the study area (region of interest), acquisition date range relevant to the analysis period, and cloud cover thresholds to ensure data quality.

Multiple spectral bands were used in the analysis, including visible, near infrared, and shortwave infrared bands, which provide useful spectral signatures for distinguishing between landcover classes such as water, sand, vegetation, and artificial surfaces.

In addition to imagery, manually digitised training polygons were created within GEE to represent representative samples of each landcover class. These polygons formed the basis of the supervised machine learning approach used in this workflow.

3.2 PROCESSING ENVIRONMENT

The workflow was implemented using a combination of cloud-based and local processing tools to enable efficient geospatial analysis and machine learning model development.

Google Earth Engine was used for accessing Sentinel-2 imagery, filtering data by location, date, and cloud cover, applying cloud masking techniques, digitising training polygons, and sampling pixel values from labelled regions. Google Earth Engine is widely used for large-scale geospatial analysis due to its ability to process satellite data in a cloud computing.

Python was used as the primary programming environment for data handling and preprocessing, machine learning model development and training, geospatial processing and analysis, and visualisation of results. The machine learning component was implemented using PyTorch, a deep learning library that supports the development and training of Artificial Neural Networks (Paszke et al., 2019).

Additional Python libraries were used to support the workflow:

- NumPy and Pandas for numerical computation and data manipulation
- Matplotlib for visualisation of results
- Optional geospatial libraries (e.g., GeoPandas, Rasterio) for handling spatial data formats

The combination of Google Earth Engine and Python enabled an integrated workflow where data acquisition, preprocessing, model training, and shoreline extraction could be performed in a consistent and reproducible manner.

3.3 IMAGE PREPROCESSING

Sentinel-2 imagery was retrieved from Google Earth Engine and filtered according to the defined region of interest, acquisition date range, and cloud cover thresholds. To ensure consistency in the dataset, only scenes meeting acceptable cloud cover conditions were retained for further processing.

Cloud contamination was addressed using the quality assessment (QA) bands provided with Sentinel-2 imagery. Cloud and cloud-shadow pixels were identified and removed using bitmasking techniques, resulting in cloud-masked images suitable for analysis (Figure 1). This step is essential in coastal environments, where cloud cover can obscure shoreline features and introduce uncertainty into classification results. The cloud masking approach follows established remote sensing practices for improving the reliability of optical imagery (Zhu *et al.,* 2015).

Figure 1:Original Sentinel-2 images (left) and the corresponding filtered one after cloud-masking is applied (right). Data source: Sentinel-2 imagery (ESA, 2025), accessed via Google Earth Engine. Available at: https://dataspace.copernicus.eu/ [Accessed 09/12/2025].

3.4 TRAINING DATA PREPRATION

Training data was generated through manual digitisation of representative landcover classes within Google Earth Engine. Polygons were created to represent surface types relevant to shoreline classification, including water, sand, agricultural land, and artificial surfaces (Figure 2).

Polygons are carefully selected to capture spatial variability across the study area and to ensure that each class was well represented. To improve the robustness of the dataset, samples were extracted from multiple images across different dates, allowing the training data to incorporate variations in seasonal conditions, and environmental factors.

Pixel values corresponding to each digitised polygon were then sampled from the multispectral imagery. These samples formed the labelled dataset used to train the supervised classification model. By combining spatially distributed training areas with multi-temporal sampling, the dataset provides a representative foundation for distinguishing between landcover classes based on their spectral characteristics.

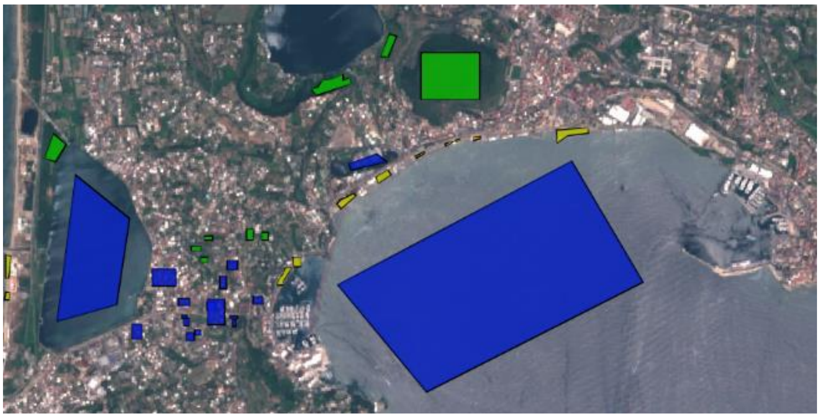

Figure 2: User-defined training polygons. Data source: Copernicus (2025) *Sentinel-2 data*. Data source: Sentinel-2 imagery (ESA, 2025), accessed via Google Earth Engine Available at: https://dataspace.copernicus.eu/ [Accessed 09/12/2025].

3.5 MACHINE LEARNING CLASSIFICATION

An Artificial Neural Network (Figure 3) was developed to perform supervised classification of the multispectral imagery. The model uses spectral input features derived from the Sentinel-2 bands to classify each pixel into one of the predefined landcover categories. Neural networks are particularly well suited to this task due to their ability to model complex, non-linear relationships between input features and class labels (Goodfellow *et al.*, 2016).

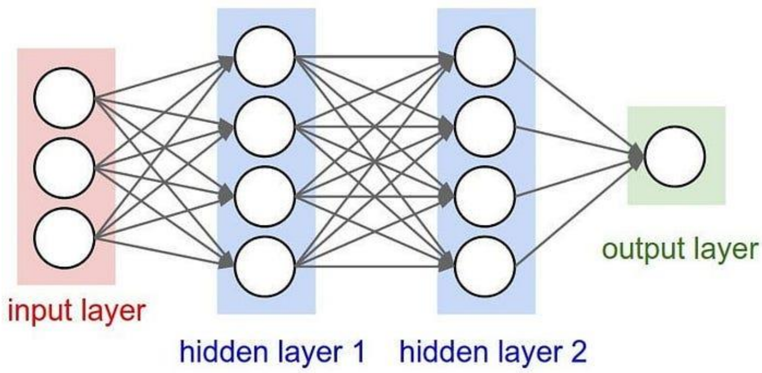

Figure 3: Structure of the ANN. Source: Medium article by Jeslur Rahman (2024). Available at: https://medium.com/@jeslurrahman/simply-understanding-artificial-neural-networks-ann-deep-learning-10b19e2eadca [Accessed 09/12/2025].

The ANN was trained using the labelled dataset generated during the training data preparation stage. During training, a weighted loss function was applied to account for class imbalance, ensuring that less frequently occurring classes were not underrepresented in the model learning process. This improves classification performance across all landcover types and reduces bias toward dominant classes such as water or vegetation.

Once trained, the model was applied to the input imagery to generate classified outputs, where each pixel is assigned a landcover label based on its spectral characteristics (Figure 4). The resulting classification maps provide the basis for subsequent shoreline extraction.

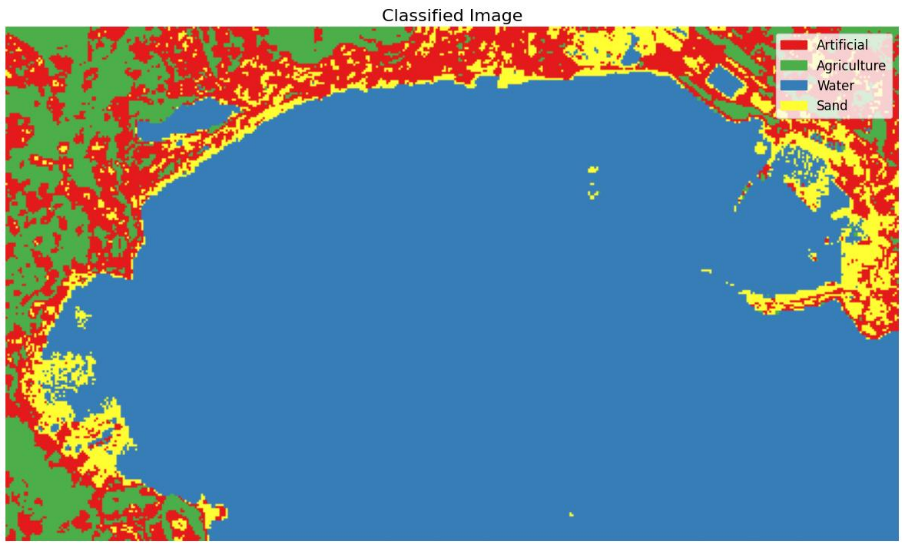

Figure 4: Four-class landcover classification showing artificial, agricultural/natural, water, and sand classes derived from Sentinel-2 imagery. Data source: Sentinel-2 imagery (ESA, 2025), accessed via Google Earth Engine. Available at: https://dataspace.copernicus.eu/ [Accessed 09/12/2025].

3.6 SHORELINE EXTRACTION

Following classification, the output landcover maps were simplified into a binary land–water representation (Figure 5). This was achieved by grouping relevant landcover classes into either land or water categories, allowing for a clear delineation of the shoreline as the boundary between these two classes.

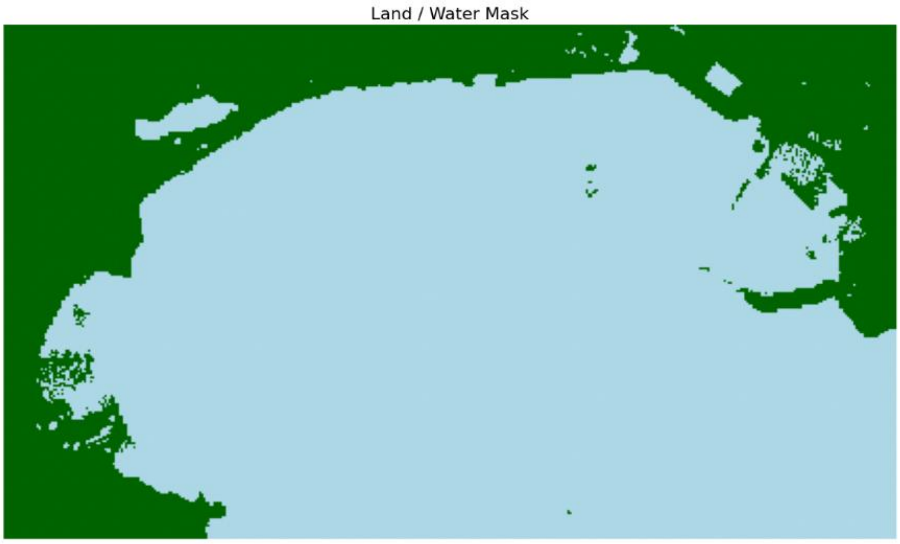

Figure 5: Corresponding binary land-water mask generated by merging non-water classes into a single land category. Data source: Sentinel-2 imagery (ESA, 2025), accessed via Google Earth Engine. Available at: https://dataspace.copernicus.eu/ [Accessed 09/12/2025].

To improve the quality of the extracted shoreline, post-processing techniques were applied to remove noise and small classification artefacts. These steps included smoothing and filtering operations designed to produce a continuous and coherent shoreline representation. The cleaned binary mask was then used to derive shoreline contours, which define the spatial position of the coastline at each time step.

The resulting shoreline vectors provide a consistent basis for comparison across multiple dates, enabling the assessment of temporal changes in shoreline position (Figure 6).

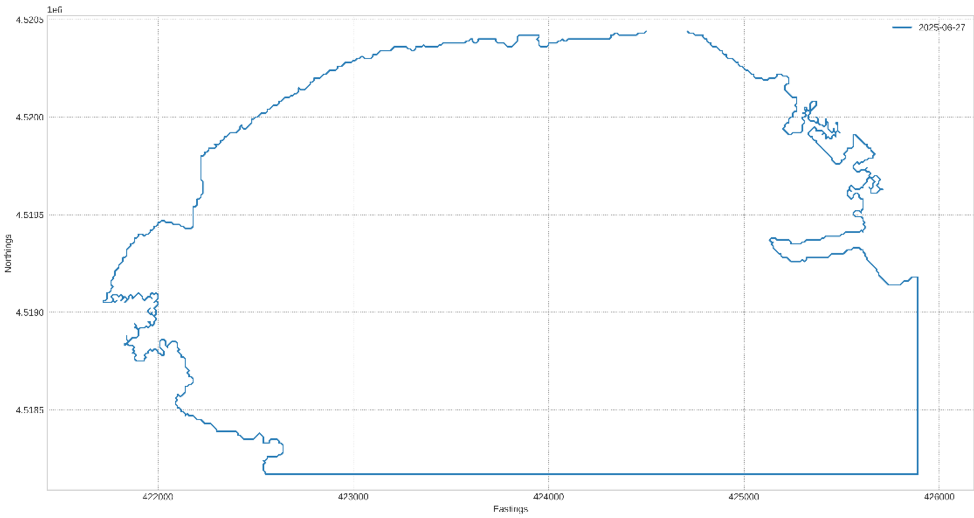

Figure 6: Final shoreline extracted from the land–water boundary, representing the coastal interface used for subsequent analysis. Data source: Copernicus Sentinel-2 imagery (European Space Agency, 2025), accessed via Google Earth Engine. Available at: https://dataspace.copernicus.eu/ [Accessed 09/12/2025].

3.7 TRANSECT BASED ANALYSIS

To quantify shoreline change over time, a transect-based analysis approach was implemented. A reference baseline was first defined parallel to the general orientation of the coastline. From this baseline, a series of shore-normal transects were generated at regular intervals along the study area.

Each transect serves as a measurement line along which shoreline positions from different time periods are intersected. The intersection points between the shoreline and each transect are used to calculate cross-shore distances, allowing for the quantification of shoreline movement over time at discrete spatial locations (Figure 7).

By comparing shoreline positions across multiple dates along each transect, rates of erosion and accretion can be derived. This approach enables both localised and spatially distributed analysis of coastal change, providing insight into patterns of shoreline variability across the study area (Figure 8).

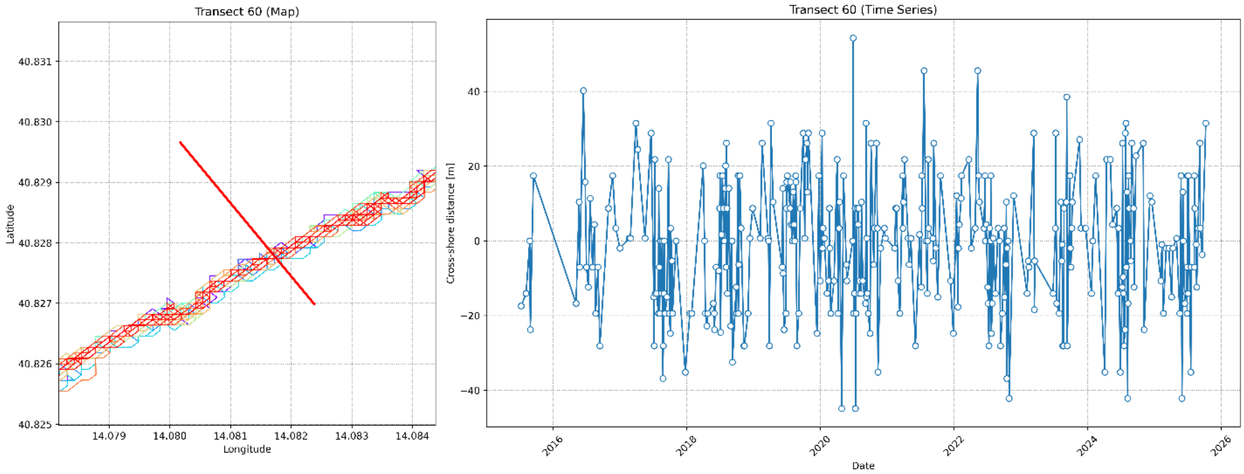

Figure 7: Shoreline intersections along a single transect showing temporal variation in shoreline position. The corresponding distance measurements are used to generate a shoreline change time series, illustrating how transect-based sampling enables quantification of coastal movement over time. Data source: Sentinel-2 imagery (ESA, 2025), accessed via Google Earth Engine. Available at: https://dataspace.copernicus.eu/ [Accessed 09/12/2025].

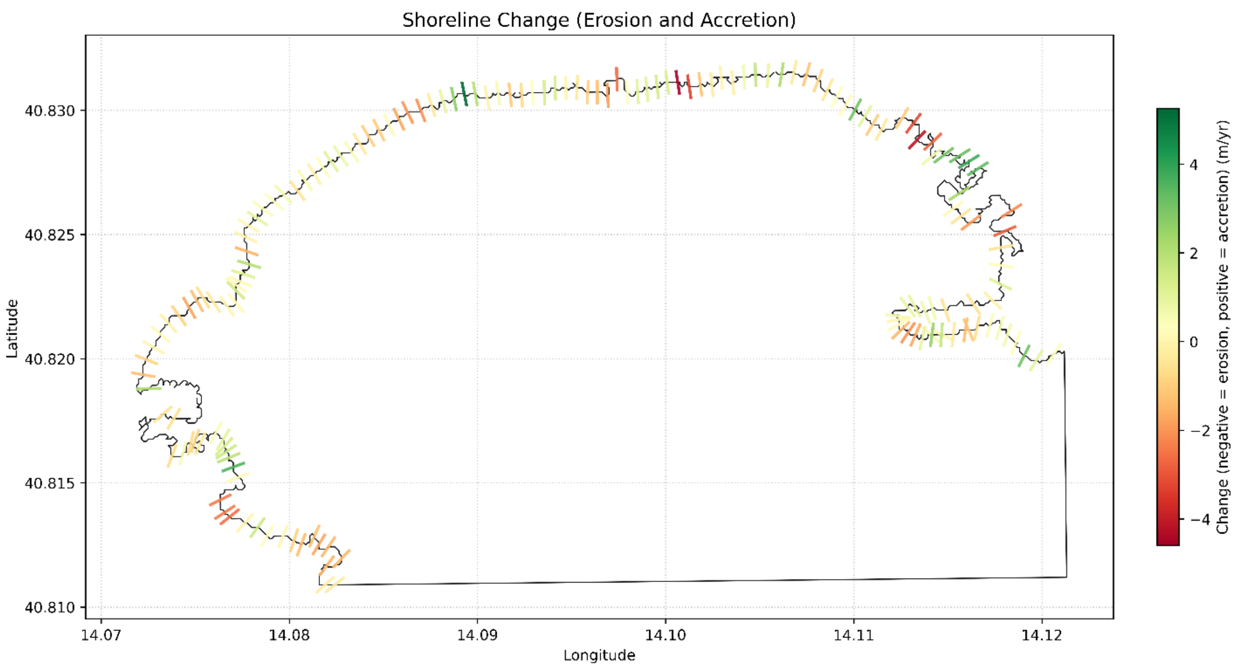

Figure 8: Spatial distribution of shoreline change across the study area derived from transect-based measurements. Transects intersect multiple shoreline positions over time, enabling calculation of erosion and accretion rates along the coast. Data source: Sentinel-2 imagery (ESA, 2025), accessed via Google Earth Engine. Available at: https://dataspace.copernicus.eu/ [Accessed 09/12/2025].

3.8 TIDAL CORRECTIONS OF SHORELINES

The shoreline positions extracted within this workflow represent the instantaneous land–water boundary at the time of satellite image acquisition. As Sentinel-2 imagery is collected at different stages of the tidal cycle, the extracted shoreline positions are influenced not only by long-term coastal change processes, but also by short-term tidal fluctuations. Consequently, the shoreline datasets generated by this workflow are not tidally corrected.

Tidal variation can introduce significant horizontal displacement in shoreline position, particularly within low-gradient coastal environments such as sandy beaches, tidal flats, estuaries, and barrier systems. In these environments, small changes in water level may correspond to large cross-shore shifts in the apparent shoreline location. If uncorrected, tidal effects may obscure or exaggerate true patterns of erosion and accretion and introduce uncertainty into shoreline change analyses (Boak & Turner, 2005).

For visualisation and qualitative interpretation, the extracted shorelines may still provide useful insight into general coastal behaviour. However, for quantitative shoreline change analysis, including the calculation of shoreline movement rates, erosion/accretion statistics, or comparison between different acquisition dates, tidal correction is recommended prior to analysis.

Tidal correction involves adjusting shoreline positions to a common vertical tidal datum or reference water level. This is typically achieved by:

- Obtaining tidal elevation data corresponding to the acquisition time of each Sentinel-2 image
- Estimating local beach slope or coastal gradient
- Calculating the horizontal shoreline offset associated with the tidal elevation difference
- Translating shoreline positions to a consistent tidal reference level (e.g. Mean Sea Level or Mean High Water)

The magnitude of tidal correction required will vary depending on local tidal range, coastal morphology, and shoreline slope. In macrotidal environments or low-gradient coastal systems, tidal correction may substantially improve the reliability of shoreline change measurements.

Users intending to undertake quantitative coastal change analysis should therefore apply an appropriate tidal correction methodology before generating shoreline change statistics or interpreting long-term trends. The extracted shorelines provided by this workflow should be considered preliminary shoreline indicators until tidal correction has been completed.

Appendix 1

INTRODUCTION

This user guide describes a workflow for extracting shoreline positions from Sentinel-2 satellite imagery using machine learning. The process enables the generation of time-series datasets to analyse coastal change over time. The workflow combines cloud-based geospatial data processing with Python-based machine learning to automate shoreline detection and reduce reliance on manual digitisation.

HOW TO USE THIS GUIDE

This workflow should be followed sequentially, as each stage builds upon the outputs of the previous step. Users should begin with training data preparation, followed by model training, image classification, shoreline extraction, transect generation, and finally shoreline change analysis. Intermediate outputs should be reviewed at each stage to ensure correctness before proceeding.

REQUIREMENTS

To run this workflow, the following are required:

- Google Earth Engine (GEE) account for image access and preprocessing (https://code.earthengine.google.com/)
- Python environment (e.g., Anaconda, Jupyter Notebook, Google Collaboratory)
- Required Python libraries:
    - PyTorch (machine learning)
    - NumPy and Pandas (data handling)
    - Matplotlib (visualisation)
    - GeoPandas / Rasterio (geospatial processing)

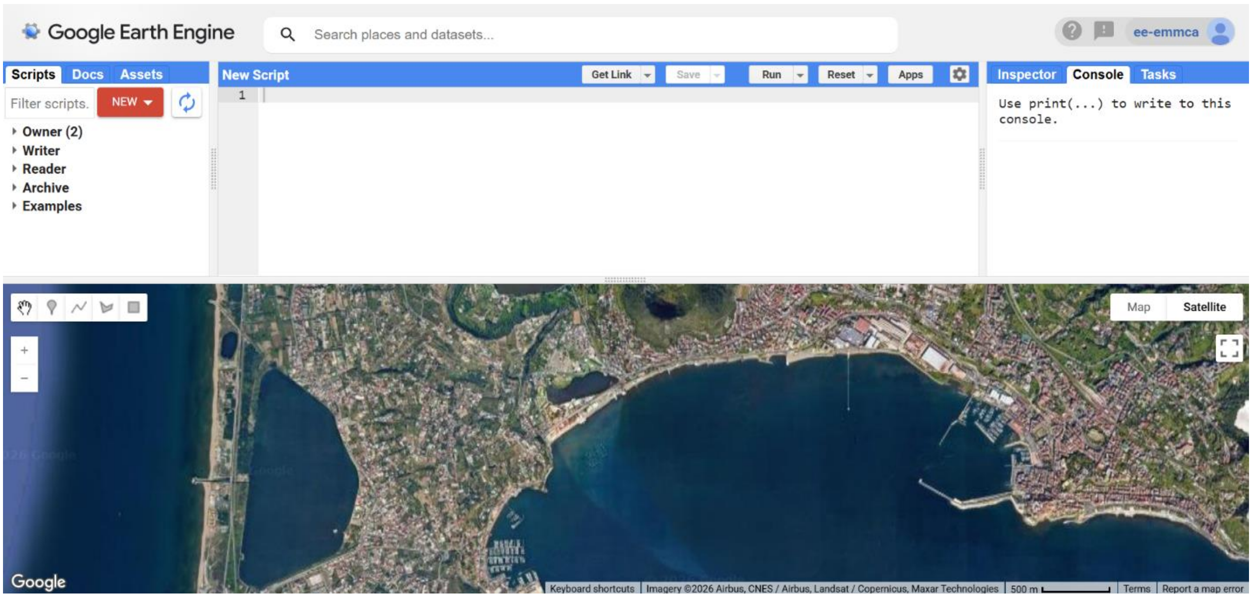

Figure 9: Screenshot of the Google Earth Engine (GEE) Code Editor interface used for accessing, visualising, and processing geospatial datasets. Code Editor available at: https://code.earthengine.google.com/

WORKFLOW OVERVIEW

The workflow consists of six main stages:

1. Preparation of training data
2. Training of the machine learning model
3. Classification of Sentinel-2 imagery
4. Shoreline extraction
5. Transect generation
6. Measurement and visualisation of shoreline change

STEP BY STEP WORKFLOW

Prepare Training Data

Training data is required to enable the supervised machine learning model to distinguish between land cover classes based on their spectral characteristics. This stage involves generating labelled samples that will be used to train and validate the model.

1. Digitise polygons in Google Earth Engine representing:
    - Water
    - Sand
    - Agricultural/Natural land
    - Artificial surfaces

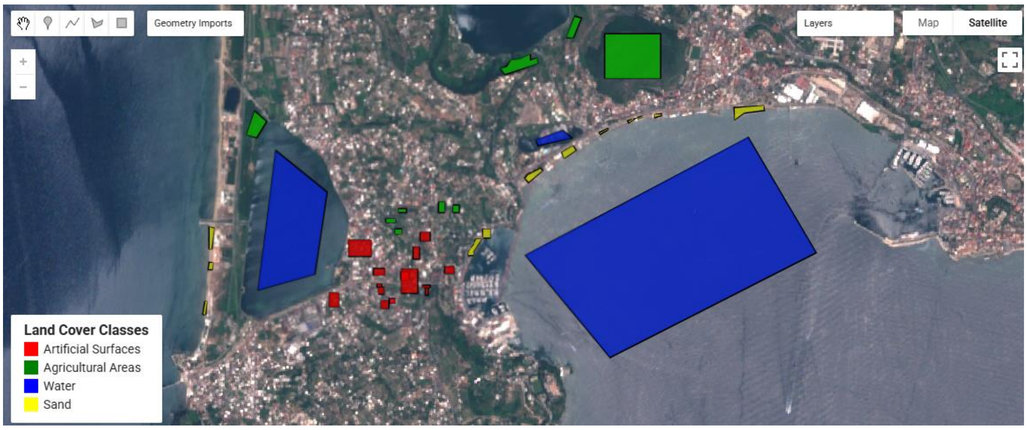

Figure 10: User-defined training polygons used for supervised classification of Sentinel-2 imagery into water, sand, agricultural/natural land, and artificial surfaces Data source: Sentinel-2 imagery (ESA, 2025), accessed via Google Earth Engine.

Ensure polygons represent homogeneous areas and cover different parts of the study area.

2. Load Sentinel-2 imagery:
    - Filter by region of interest (ROI)
    - Filter by date range
    - Apply a cloud cover threshold (e.g., <10%)

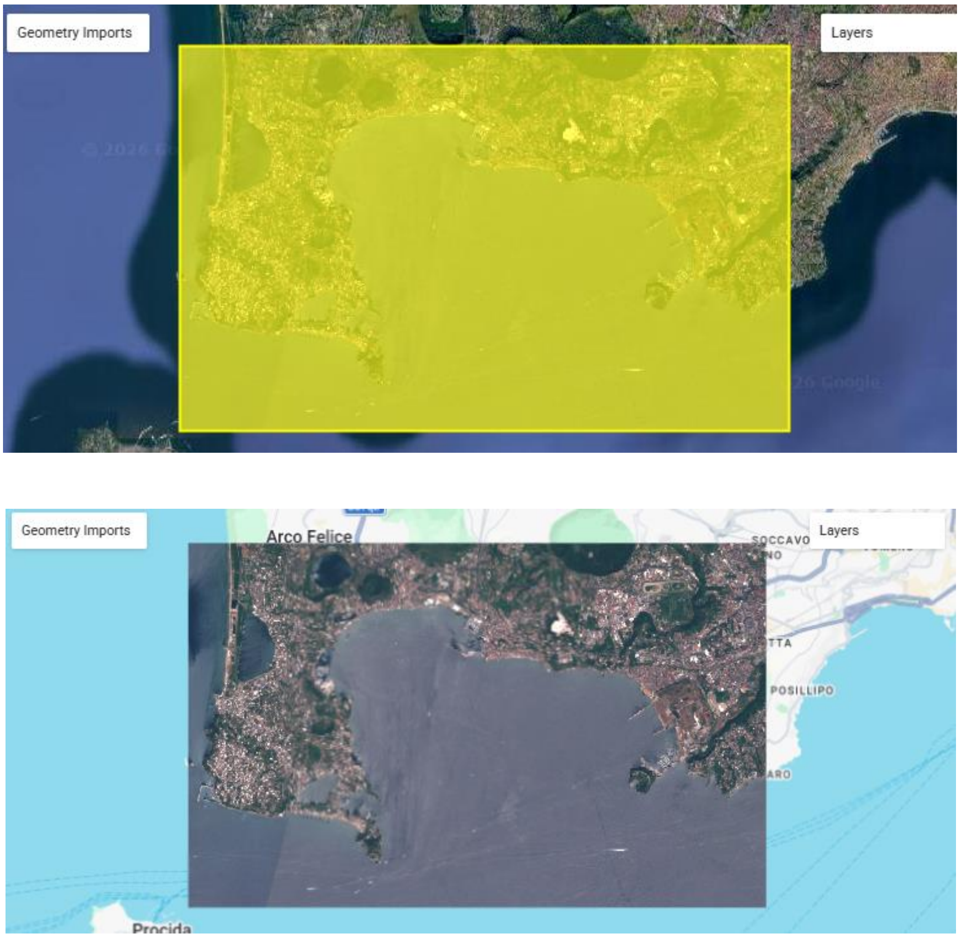

Figure 11: Google Earth Engine (GEE) Code Editor interface showing Sentinel-2 imagery with a user-defined region of interest (ROI) polygon (top) and the corresponding image clipped to the ROI boundary (bottom). Data source: Sentinel-2 imagery (ESA, 2025), accessed via Google Earth Engine. Code Editor available at: https://code.earthengine.google.com/ [Accessed 09/12/2025].

3. Apply cloud masking:
    - Remove pixels affected by clouds and cirrus
    - Retain only valid (cloud-free) observations for analysis
4. Extract spectral pixel values:
    - Use relevant Sentinel-2 bands (e.g., B2, B3, B4, B5, B8, B11)
    - Sample pixels within each training polygon
5. Combine all samples into a single dataset:
    - Merge all class samples into one structured dataset
    - Export as a CSV file containing input features (spectral bands) and corresponding class labels
6. Split the dataset into:
    - Training set for model fitting
    - Validation set for hyperparameter tuning and monitoring performance
    - Test set for independent evaluation of model accuracy

landcover	B2	B3	B4	B5	B8	B11
0.0	0.1521	0.1398	0.1595	0.1941	0.1707	0.2685
0.0	0.1475	0.1484	0.1658	0.1941	0.2028	0.2685
0.0	0.1731	0.168	0.227	0.199	0.2648	0.2894
1.0	0.0901	0.0882	0.0563	0.1113	0.3197	0.1678
1.0	0.0961	0.0974	0.0739	0.1113	0.3129	0.1678
1.0	0.1215	0.1246	0.1128	0.1325	0.3052	0.1942
2.0	0.1132	0.0915	0.0668	0.0593	0.0568	0.0414
2.0	0.1125	0.0909	0.068	0.06	0.0573	0.0429
2.0	0.1137	0.0908	0.0688	0.06	0.0583	0.0429
3.0	0.1389	0.1283	0.1417	0.1577	0.1789	0.2151
3.0	0.147	0.1464	0.1662	0.1577	0.1842	0.2151
3.0	0.1566	0.1492	0.1816	0.1916	0.1919	0.2429

Figure 12: Example of the training dataset, containing Sentinel-2 spectral band values as input features and corresponding land cover class labels. Each row represents an individual pixel sample extracted from digitised training polygons and used to train the classification model. Data source: Sentinel-2 imagery (ESA, 2025), accessed via Google Earth Engine.

Train the Machine Learning Model

This step describes how the Artificial Neural Network (ANN) is defined, trained, and evaluated using the prepared training dataset. The model learns to classify pixels into land cover classes based on spectral information extracted from Sentinel-2 imagery.

1. Define the model:
    - Specify the ANN architecture (e.g. number of layers, neurons per layer if known)
    - Define activation functions (e.g. ReLU for hidden layers, softmax for output)
    - Output layer corresponds to four land cover classes (water, sand, vegetation, artificial)
2. Set training parameters:
    - Learning rate (controls update step size during optimisation)
    - Batch size (defines number of samples processed per iteration)
    - Number of epochs (determines how many times the full dataset is used for training)
3. Apply class weighting:
    - Weights are computed based on class frequency in the training dataset
        - Assign higher weights to underrepresented classes (e.g., sand)
        - Helps reduce bias toward dominant classes (e.g., water)
4. Train the model:
    - Training data is fed into the model in batches
    - The model iteratively learns relationships between spectral inputs and class labels
    - Internal weights are adjusted using backpropagation
    - Loss and accuracy metrics are tracked to monitor convergence and overall performance.

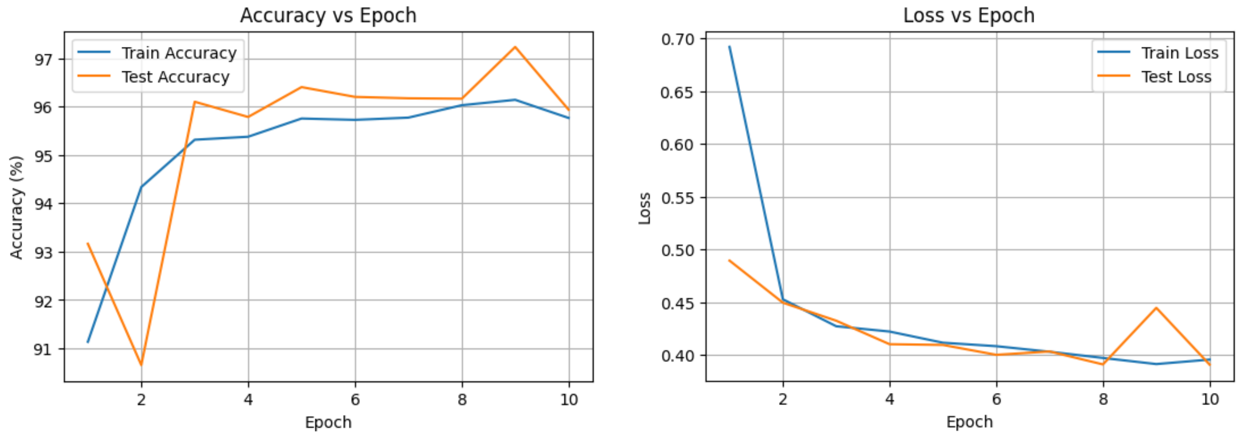

Figure 13: Training performance of the Artificial Neural Network showing (left) model accuracy and (right) model loss as a function of training epochs. The plots illustrate model convergence, with accuracy increasing and loss decreasing over successive epochs as the network learns from the training dataset.

5. Evaluate performance:
    - Test the model on validation dataset
    - Generate a confusion matrix to assess classification performance
    - Identify misclassification patterns to identify weaknesses in the model

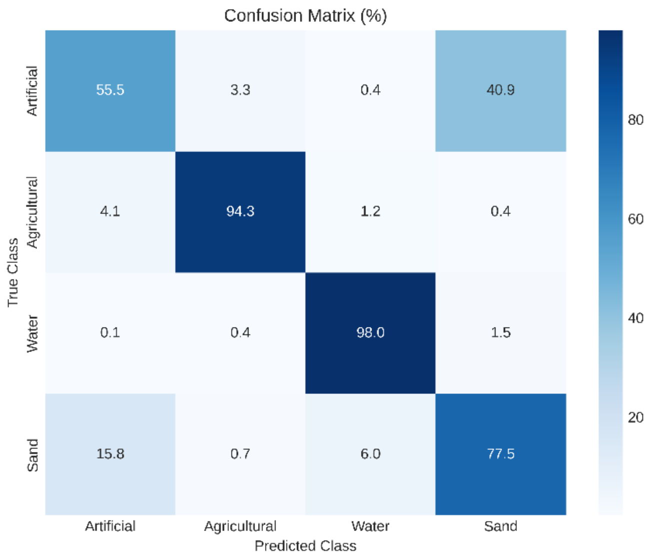

Figure 14: Confusion matrix (%) illustrating the classification performance of the Artificial Neural Network for Artificial, Agricultural/Natural, Water, and Sand classes, with diagonal values indicating correct classifications and off-diagonal values representing misclassifications.

Classify Sentinel-2 Imagery

The trained Artificial Neural Network is applied to Sentinel-2 imagery to generate a pixel-wise land cover classification. Each pixel is assigned a class label based on its spectral features, producing a classified raster representing spatial land cover distribution.

- Load Sentinel-2 images for the study area
- Select required spectral bands
- Convert imagery into array format suitable for model input
- Apply the trained ANN to classify each pixel

The output is pixel-level land cover classification map representing the spatial distribution of classes across the study area.

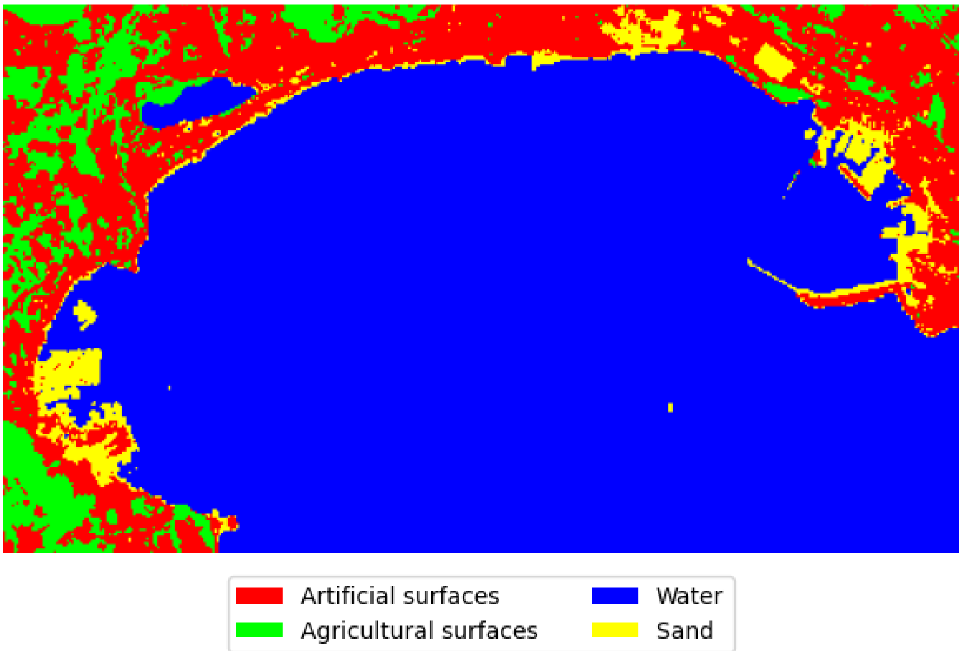

Figure 15: Four-class land cover classification map showing artificial, agricultural/natural, water, and sand classes derived from Sentinel-2 imagery using an Artificial Neural Network. Data source: Sentinel-2 imagery (ESA, 2025), accessed via Google Earth Engine. Available at: https://dataspace.copernicus.eu/ [Accessed 09/12/2025].

Extract Shorelines

Shorelines are derived from the classified land cover output by isolating the boundary between land and water classes and converting it into vector form.

1. Convert classification into a binary land-water mask:
    - Water pixels are assigned a value of 1
    - All other classes (e.g. land) are assigned a value of 0

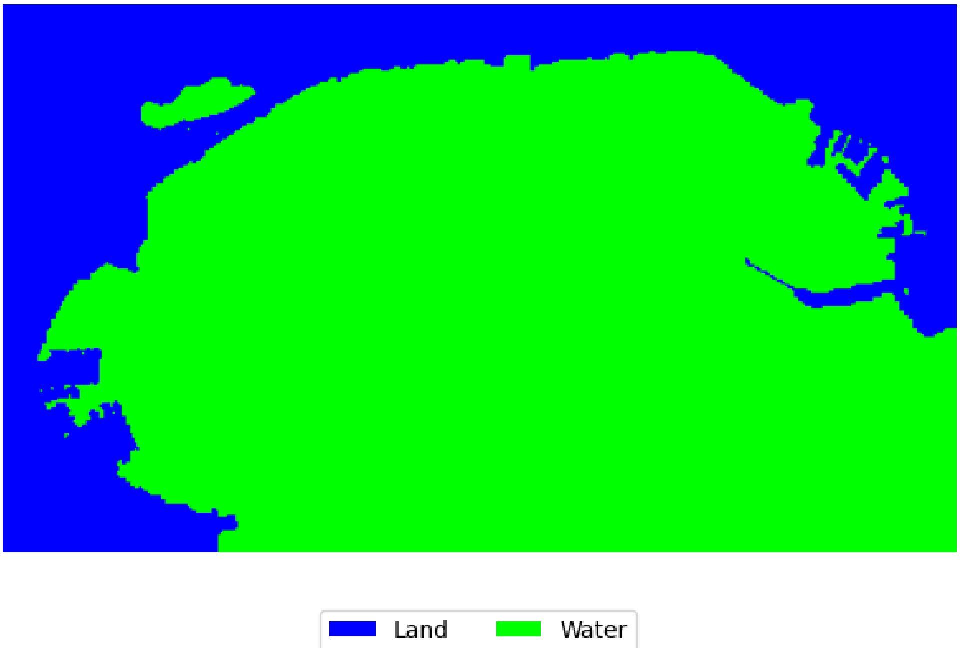

Figure 16: Binary land-water mask derived from Sentinel-2 imagery by converting the multi-class classification into a two-class output, where water pixels are assigned a value of 1 and all other classes (e.g. land) are assigned a value of 0 Data source: Sentinel-2 imagery (ESA, 2025), accessed via Google Earth Engine Available at: https://dataspace.copernicus.eu/ [Accessed 09/12/2025].

2. Apply post-processing:
    - Remove small, isolated regions (noise)
    - Apply morphological filtering to smooth the binary mask and improve boundary definition
3. Extract contours:
    - Identify boundaries between land and water
    - Convert boundaries into vector shoreline lines
4. Filter results:
    - Remove short or fragmented contours
    - Retain only primary shoreline features

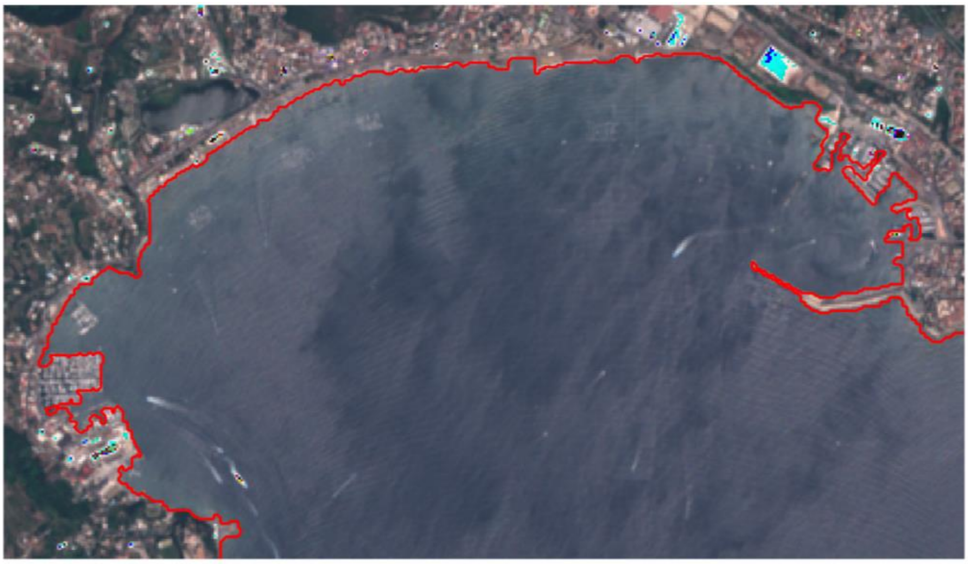

Figure 17: Sentinel-2 imagery with the extracted shoreline overlaid, derived from Sentinel-2 data. Data source: Sentinel-2 imagery (ESA, 2025), accessed via Google Earth Engine. Available at: https://dataspace.copernicus.eu/ [Accessed 09/12/2025].

Generate Transects

Transects are generated to enable consistent measurement of shoreline change over time by providing fixed cross-shore sampling lines.

1. Define a reference line along the coastline
    - Create a baseline that follows the orientation of the coastline
2. Generate points along the reference line:
    - Sample points at regular intervals (e.g., 50-100 m)
3. Generate transects at each point:
    - Calculate the local orientation of the coastline
    - Create a perpendicular (shore-normal) transect
4. Define transect length:
    - Ensure transects are long enough to intersect all shoreline positions across the dataset.
5. Store transects for analysis
    - Save transects as vector line geometries for use in shoreline change analysis

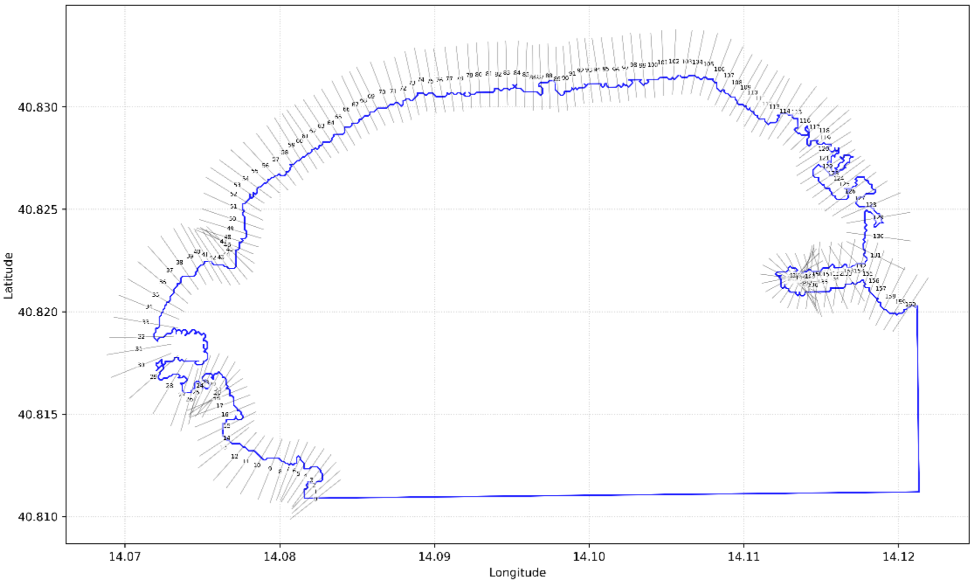

Figure 18: Baseline and shore-normal transects generated along the coastline from Sentinel-2 imagery for shoreline analysis. Data source: Copernicus Sentinel-2 imagery (European Space Agency, 2025), accessed via Google Earth Engine. Available at: https://dataspace.copernicus.eu/ [Accessed 09/12/2025].

Measure Shoreline Change

Shoreline movement is quantified by analysing the intersection of shoreline positions with the generated transects over time.

1. For each transect:
    - Identify intersections with shoreline positions at each time step
    - Calculate cross-shore distances from the reference line

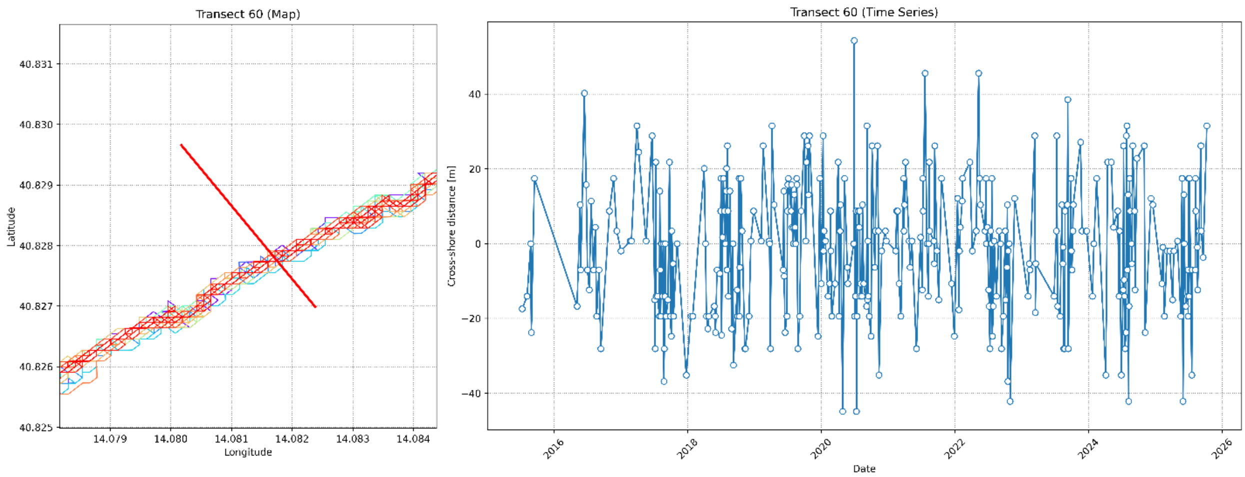

Figure 19: Shoreline intersections along a transect showing temporal variation in shoreline position, previously derived from Sentinel-2 imagery. Data source: Sentinel-2 imagery (ESA, 2025), accessed via Google Earth Engine. Available at: https://dataspace.copernicus.eu/ [Accessed 09/12/2025].

2. Compile results into a dataset:
    - Rows represent dates (time steps)
    - Columns represent transects
    - Values represent shoreline distance measurements along each transect

3. Handle multiple intersections:
    - Where multiple shoreline intersections occur along a transect, compute a representative value (e.g., median or mean distance)

This produces a time-series dataset of shoreline positions and movement along each transect, enabling analysis of erosion and accretion patterns over time.

OUTPUTS

The workflow produces the following outputs:

- Extracted shoreline coordinates representing the land-water boundary
- Binary land-water masks derived from the classified imagery
- Land cover classification maps showing spatial distribution of classes
- Transect geometries used as reference lines for analysis
- Time-series datasets quantifying shoreline change over time
- Visual plots to support interpretation, validation, and presentation of results

References

Boak, E.H. and Turner, I.L., 2005. Shoreline definition and detection: a review. *Journal of coastal research*, *21*(4), pp.688-703. https://doi.org/10.2112/03-0071.1

Drusch, M., Del Bello, U., Carlier, S., Colin, O., Fernandez, V., Gascon, F., Hoersch, B., Isola, C., Laberinti, P., Martimort, P., Meygret, A., Spoto, F., Sy, O., Marchese, F. and Bargellini, P. (2012) 'Sentinel-2: ESA's optical high-resolution mission for GMES operational services', *Remote Sensing of Environment*, 120, pp. 25–36. https://doi.org/10.1016/j.rse.2011.11.026

European Space Agency (n.d.) *Sentinel-2*. Available at: https://www.esa.int/Applications/Observing_the_Earth/Copernicus/Sentinel-2 (Accessed: 09 December 2025).

Goodfellow, I., Bengio, Y. and Courville, A. (2016). *Deep Learning*. Cambridge, MA: MIT Press. Available at: https://www.deeplearningbook.org/

Google Earth Engine (n.d.) *Google Earth Engine*. Available at: https://earthengine.google.com/ (Accessed: 09 December 2025).

Gorelick, N., Hancher, M., Dixon, M., Ilyushchenko, S., Thau, D. and Moore, R. (2017). Google Earth Engine: Planetary-scale geospatial analysis for everyone. *Remote Sensing of Environment*, 202, pp. 18–27. https://doi.org/10.1016/j.rse.2017.06.031

McAllister, E., Payo, A., Novellino, A., Dolphin, T. and Medina-Lopez, E., 2022. Multispectral satellite imagery and machine learning for the extraction of shoreline indicators. *Coastal Engineering*, *174*, p.104102. https://doi.org/10.1016/j.coastaleng.2022.104102

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L. and Desmaison, A., 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, *32*. https://doi.org/10.48550/arXiv.1912.01703

Zhu, Z., Wang, S. and Woodcock, C.E., 2015. Improvement and expansion of the Fmask algorithm: Cloud, cloud shadow, and snow detection for Landsats 4–7, 8, and Sentinel 2 images. *Remote sensing of Environment*, *159*, pp.269-277. https://doi.org/10.1016/j.rse.2014.12.014