Satellite imagery pre-processing and feature extraction for the mapping of coastal ecosystems using Google Earth Engine: A workflow for practitioners

Ahmad Badruzzaman [a], Prawesti Wulandari [a,b], Sainal Sainal [a], Matthew Ashley [c], Susan Jobling [d,e], Melanie C. Austen [c], Radisti A. Praptiwi [a,f,*]

[a] Sustainability Research Cluster and Department of Biotechnology, Universitas Esa Unggul, Jl. Arjuna Utara No. 9, Kebon Jeruk, Jakarta 11510, Indonesia
[b] Yayasan Puspa Hanuman Indonesia, Bogor, Indonesia
[c] School of Biological and Marine Sciences, University of Plymouth, Drake Circus, Plymouth, Devon PL4 8AA, United Kingdom
[d] Civil and Environmental Engineering, College of Design, Engineering and Physical Sciences, and Centre for Pollution Research and Policy, Brunel University London, Uxbridge UB8 3PH, United Kingdom
[e] PISCES Partnership, Brunel University London, Uxbridge UB8 3PH, United Kingdom
[f] Research Center for Ecology and Ethnobiology, National Research and Innovation Agency (BRIN), Jl. Raya Jakarta-Bogor KM 46, Cibinong, Bogor 16911, Indonesia

GRAPHICAL ABSTRACT
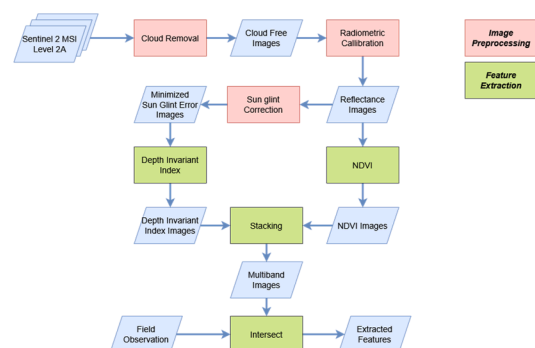
Related research article: None.

* Corresponding author.
  E-mail address: radisti_ayu@outlook.com (R.A. Praptiwi).

A B S T R A C T

The use of Google Earth Engine (GEE) is increasingly common in geospatial analysis of satellite images for various environmental management purposes due to its easy accessibility and capabilities to support complex pre-processing and mining of geographic data. In the context of coastal management, GEE provides opportunities for cost-efficient mapping of coastal habitats and their ecosystem service potentials. Understanding the extent of coastal habitats and the spatial and temporal variabilities of their ecosystem services can be useful for management and intervention purposes. GEE is well-suited for this due to its user-friendliness, particularly for non-experts of programming languages, such as area managers and other practitioners. However, there is no specific methodological guideline for the pre-processing and feature extraction of satellite images in GEE that can be readily adopted by these practitioners. This study develops general methodological steps to perform those processes that can be adapted to different management needs. Highlights of this study:

- Steps detailed in this method paper will produce processed satellite images readily applicable for machine learning to classify coastal ecosystems.
- The development of this adaptable workflow can benefit and empower local area managers, particularly in low-resource settings, to conduct monitoring of their area.

Specifications table

Subject area	Earth and Planetary Sciences
More specific subject area	GIS remote sensing, environmental monitoring.
Name of your method	Sentinel-2 Satellite Image Pre-processing and Feature Extraction for Coastal Habitat Mapping.
Name and reference of original method	J. W. Rouse, R. H. Haas, J. A. Schell and D. W. Deering. (1974) Monitoring vegetation systems in the great plains with ERTS (Earth Resources Technology Satellite).
	D. Lyzenga. (1978) Passive remote sensing techniques for mapping water depth and bottom features. https://doi.org/10.1364/AO.17.000379.
	S. McFeeters. (1996) The use of the Normalized Difference Water Index (NDWI) in the delineation of open water features. https://doi.org/10.1080/01431169608948714.
	D. R. Lyzenga, N. P. Malinas and F. J. Tanis. (2006) Multispectral bathymetry using a simple physically based algorithm. https://doi.org/10.1109/TGRS.2006.872909.
	S. Kay, J. Hedley and S. Lavender. (2009) Sun glint correction of high and low spatial resolution images of aquatic scenes: A review of methods for visible and near-infrared wavelengths. https://doi.org/10.3390/rs1040697.
	P. Wicaksono, P. Aryaguna and W. Lazuardi. (2019) Benthic habitat mapping model and cross validation using machine-learning classification algorithms. https://doi.org/10.3390/rs11111279.
	R. Martínez Prentice, M. Villoslada Peciña, R. Ward, T. Bergamo, C. Joyce and K. Sepp. (2021) Machine learning classification and accuracy assessment from high-resolution images of coastal wetlands. https://doi.org/10.3390/rs13183669.
	P. Wicaksono, S. Wulandari, W. Lazuardi and M. Munir. (2021) Sentinel-2 images deliver possibilities for accurate and consistent multi-temporal benthic habitat maps in optically shallow water. https://doi.org/10.1016/j.rsase.2021.100572.
Resource availability	Google Earth Engine; https://scihub.copernicus.eu/

Background

Coastal ecosystems are essential for the flourishing of human individuals, communities, and societies through their supply of ecosystem services, such as food provisioning, water flow regulation, shoreline protection, and recreational activities [1,2]. The types of ecosystem services and their quantities are particular to specific types of habitats, such as coral reefs and seagrass meadows, due to their inherent ecological structures and functioning [1]. Analysis of ecosystem services can thus be performed at the level of a particular habitat as a service providing unit (SPU) [3]. Therefore, monitoring of coastal ecosystems is important to understand the spatio-temporal dynamics of the ecosystem services they provide and to provide input for appropriate interventions and management of coastal areas [4,5].

In the past decade, remote sensing has increasingly been applied to map coastal ecosystems for monitoring and management purposes [6]. This growing adoption of remote sensing is spurred by its lower resource requirements compared to in-situ monitoring in coastal environment [6,5]. Despite its potential, the uptake of remote sensing applications in the context of low-resource settings still presents several challenges, mainly due to the advanced skills requirement, affordability, and the complex processes of pre-processing and mining of geographic data [5,7]. The use of Google Earth Engine (GEE) can address such challenges, owing to its capabilities in supporting various types of geospatial data and providing a suite of Earth observation data at no-cost [7]. Launched in 2010, GEE is currently the most popular cloud computing platform that provides ease of access for non-experts of programming languages and is capable of operating with common programming languages such as JavaScript and Python [8,9].

However, there is currently no specific methodological guideline on applying image preprocessing and feature extraction steps in GEE to satellite imagery of coastal environments to enable ecosystem mapping, particularly for beginner users. This condition hinders the utilization of the open-access GEE in non-academic low-resource settings, despite the potential of the platform to broaden access to remote sensing technologies and techniques beyond academia [7]. As such, this paper aims to address this knowledge gap, particularly in relation to the needs of practitioners, such as area managers, conservationists, and ecotourism operators, who often lack advanced research methodology training. Enhancing the capacity of local practitioners to address area-specific management challenges is particularly important due to the increasing decision-making responsibilities that these actors have in many coastal and marine contexts [10]. Thus, we specifically provide general methodological steps to perform satellite image pre-processing and feature extraction directly in GEE that can be adapted in various applied contexts for further spatial analysis as required by particular stakeholder needs. The methods described here are designed to enable immediate applications by practitioners in low-resource settings, especially in the Global South. However, the application of our methodological steps still requires a basic knowledge of the Python programming language. There is a variety of resources to learn the language, including from the official Python website (https://www.python.org/about/gettingstarted/) and a range of free online courses [11–13]. GEE itself can also be used in Java programming language, with an online guide for introductory users of GEE, for either Python or Java, available on the GEE website (https://developers.google.com/earth-engine/guides). As indicated in the GEE guide website, the use of Python is more appropriate for more demanding geospatial tasks, such as ecosystem classification based on large datasets, due to the larger libraries for data processing and analysis of satellite images.

Method details

Datasets acquisition

Remote sensing images

Although various satellite imagery sources can be used in GEE, in this paper we utilized the publicly available Sentinel 2 Multi-spectral Instrument (MSI) images for the purpose of illustrating the utility of our methodological approach for practitioners (Fig. 1). The Sentinel 2 satellite images should be acquired for the period of interest, according to the specified geographical area in the following website: https://scihub.copernicus.eu/. Instructions to import the obtained satellite images to GEE are available from the following online guides: https://developers.google.com/earth-engine/guides/image_upload. The images should be filtered by cloud coverage under 30 % of the scene and bounded by the study area. If the atmospheric information was unavailable, the usage of Sentinel 2 MSI images should be specified to level 2A data. The atmospheric and terrain can be corrected using the Sen2Cor algorithm with global scale data (low resolution) provided by the European Space Agency (ESA). Although the collected satellite images can be
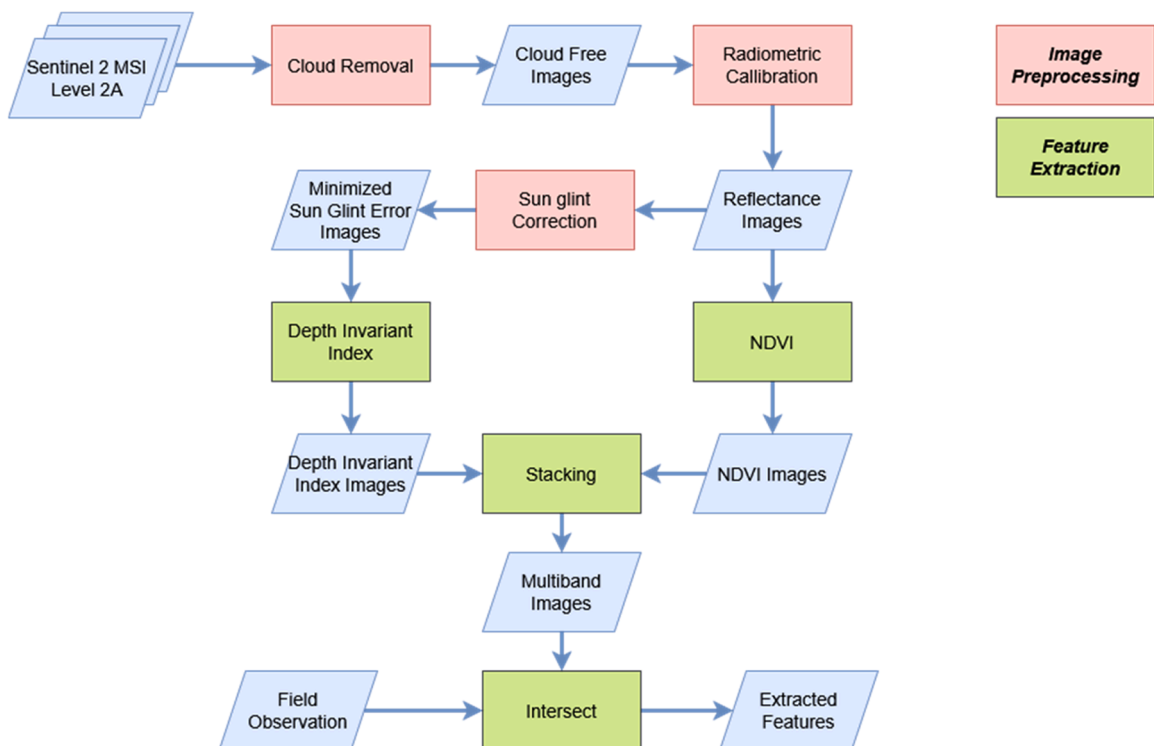

Fig. 1. Methodological steps described in this paper.

enormous, the usage of Sentinel 2 MSI level 2 data on GEE can simplify the complexity, since Sentinel 2 MSI level 2 periodically records data enabling the acquisition of spectral value under cloud cover from different observation periods with atmospheric and terrain pre-corrections. In GEE, the collection of remote sensing images can be accessed using the function of "ee.ImageCollection", with the codes shown in Fig. 2.

Field observation

If field observation data, such as coordinates of habitats, is needed for further analysis, this should be based on the commonly used geometrical observation, i.e. point, polyline, and polygon. In general, geotagging must be conducted on the node of every geometrical observation while considering the geospatial resolution of satellite images. Specific to the point observation, the observational classes should be determined by the majority of objects within the geometric resolution of the satellite images. The field observation data uploaded to GEE can be identified as features, using the function of "ee.FeatureCollection" to access the field observation data, with the codes shown in Fig. 3. If the field observation data have nodes exceeding the GEE limit, separate Comma Separated Values (CSV) files for each observational class should be prepared before uploading to the GEE asset (for instance in Excel this can be done by using the concatenate function), which can then be accessed using the codes in Fig. 3.

Image preprocessing

Direct use of raw satellite images for habitat mapping is not recommended due to distortions or noise that may be present in the raw images as a result of geometric distortions [14]. As noise and error are inevitable in any remote sensing data, image preprocessing must be performed to account for them. Figs. 4 and 5 detail the codes to conduct image preprocessing in GEE, further specified in the following separate subsections detailing the rationale behind the GEE codes. The selection of methods arranged in this workflow follows the well-established and commonly used approaches for application in benthic habitat mapping.

Cloud removal

Optical remote sensing images are rarely free of clouds, despite the pre-application of the filtering function. Cloud cover and associated shadows could obscure the surface features, hence impacting the analysis accuracy. In Sentinel 2 MSI imagery, cloud and cirrus are recorded in the QA60 Band. Therefore, the cloud mask should be computed by using this band to remove the cloud and cirrus. An alternative approach can be performed by using the MSK_CLDPRB band to select pixels with 0 % probability on both cloud and cirrus. Depending on the analytical needs, combining these approaches can be complementary.

The collected satellite images themselves can compensate for cloud coverage through their combined analysis. However, several regions of the images may overlap and become redundant due to the different satellite observation times. Therefore, statistical computation is required to reduce image collection requirements and enable a single observational image. For this purpose, combining satellite images of the same region should be performed by assuming that the observed reflectance data might be skewed due to abnormal natural phenomena, which act as outliers, hence median values should be used (outlier). Furthermore, this image reduction can be performed either after pre-processing or simultaneously with radiometric correction (described below) to minimize error in this phase, and with codes to do so in GEE displayed in Fig. 4.

Radiometric correction

The pixel values of remote sensing images are available in bit format, as Sentinel 2 MSI imagery uses 12-bit values representing surface reflectance, despite GEE relying on the reflectance format. Thus, radiometric correction should be used to convert the pixel values from the bit format to the reflectance format. Let ρ_i as pixel value of ith band on bit format and ρ_i' as radiometric corrected pixel value of ith band. Mathematically, radiometric correction for Sentinel 2 MSI imagery can be notated as follows, with the codes to do so in GEE displayed in Fig. 4:

$$\rho_i' = \rho_i \div 10000$$

Sun glint correction

Coastal environment usually has a wide surface area of water that remote sensing observation can be significantly influenced by sunlight reflection. Therefore, sun glint correction is necessary to correct the specular reflection, hence improving image interpret-

```
# Assign a variable to access collection of remote sensing images

variable = ee.ImageCollection('Remote sensing data ID')


# Filter remote sensing data by cloud

Variable = variable.filter(ee.Filter.lt('CLOUDY_PIXEL_PERCENTAGE', 30))

# Filter remote sensing data by date

Variable = variable.filterDate('start date', 'end date')

# Filter remote sensing data by area

Variable = variable.filterBounds(variable_area)
```

Fig. 2. Pseudocode for accessing satellite image collection on Google Earth Engine.

```
# Assign a variable to access asset of field observation

Variable_A = ee.FeatureCollection('Asset ID')

Variable_B = ee.FeatureCollection('Asset ID')


# In case for merge partition of field observation

Combine_var = variable_A.merge(variable_B)
```

Fig. 3. Pseudocode for accessing field observation as feature collection on GEE.

```
# Function of cloud removal and radiometric
# correction for Sentinel 2 MSI
def preprocess(image):
    scl = image.select('SCL')
    qa = image.select('QA60')
    cloudProb = image.select('MSK_CLDPRB')


    # Bits 10 and 11 are clouds and cirrus from QA60 Band, respectively
    cloud   = int(2**10)
    cirrus  = int(2**11)


    # Selecting pixels with 0% Cloud Probability and Cover Prediction of
    # Cirrus and Shadow from SCL Band
    cl_prob = cloudProb.eq(0)
    cir_prob = scl.eq(10).And(scl.eq(9)).And(scl.eq(8)).And(scl.eq(7))
    shadow = scl.eq(3)


    # Both flags should be set to zero, indicating clear conditions
    masked = qa.bitwiseAnd(cloud).eq(0).And(qa.bitwiseAnd(cirrus).eq(0))
    mask = cl_prob.And(cir_prob.neq(1)).And(shadow.neq(1))


    return image.updateMask(masked).updateMask(mask).divide(10000)
```

Fig. 4. Pseudocode for cloud removal and radiometric correction as proposed scheme on GEE.

ability and preserving spectral information. Sun glint correction can be performed by using a common sun glint correction method [15, 16] that corrects the true color band (RGB) using NIR band. Let L_i' as sun glint corrected of the ith band, Li as pixel value of the ith band, L_{NIR} as pixel value of NIR band, and $\overline{L}_{NIR}$ as the mean value of the observed NIR pixel value on the scene. Mathematically, sun glint correction can be notated as follows, with the GEE codes displayed in Fig. 5.

$$r_{ij} = \frac{Cov(i, NIR)}{Var(NIR, NIR)}$$

$$L_i' = L_i - r_{ij}[L_{NIR} - \overline{L}_{NIR}]$$

```
# Function of sun glint correction as proposed by Lyzenga et. al. (2006)
def sun_glint(image):
  bands = ['B2','B3','B4']
  mean = ee.Reducer.mean()
  cov = ee.Reducer.covariance()
  nir = image.select('B8')
  nir_hat  = nir.reduceRegion(reducer=mean,geometry=bound).get('B8')


  for i in bands:
     arr       = image.select([i,'B8']).toArray()
     varcov    = ee.Array(arr.reduceRegion(reducer=cov,geometry=bound).get('array'))
     corr      = nir.subtract(ee.Image(nir_hat))
     rho       = ee.Image(varcov.get([0,1])).divide(ee.Image(varcov.get([1,1])))
     img_corr = image.select(i).subtract(corr.multiply(rho))


     image.addBands(img_corr, None, True)


  return image
```

Fig. 5. Sun glint correction.

Feature extraction

The observed features in coastal areas of interest are derived from satellite images. These features are calculated using band math calculation. The following subsections describe the process of doing so, by selecting features that can be used for further analysis, such as the classification of coastal ecosystems based on machine learning [17,18].

Depth invariant bottom index

The reflectance value of coastal features recorded by sensors may be altered due to the effect of the water column. This can induce spectral inconsistencies which may result in misclassification of coastal ecosystems. Depth Invariant Index (DII) can be used to avoid this problem (DII). The method to obtain DII normalizes the effect of water column energy attenuation with the ratio algorithm [19, 20]]. Let ρ_r as the reflectance of the red band, ρ_g as the reflectance of the green band, and ρ_b as reflectance of the blue band. Mathematically, DII can be notated as follows:

$$DII_{12} = \frac{\rho_b}{\rho_g}$$

$$DII_{13} = \frac{\rho_b}{\rho_r}$$

$$DII_{23} = \frac{\rho_g}{\rho_r}$$

```
# Assign a function to compute depth invariant bottom index for
# each image on image collection
def water_corr(image):
   dii12 = image.select('B2').divide(image.select('B3')).rename('D12')
   dii13 = image.select('B2').divide(image.select('B4')).rename('D13')
   dii23 = image.select('B3').divide(image.select('B4')).rename('D23')

   return image.addBands(dii12, None, True)
        .addBands(dii13, None, True).addBands(dii23, None, True)
```

Fig. 6. Depth invariant bottom index.

In GEE, DII features are computed using the function of "ee.Image.divide" as shown in Fig. 6.

Normalized difference vegetation index (NDVI)

Observational classes of coastal ecosystems often contain chlorophyll from planktonic communities and vegetation, such as mangroves and seagrass [21–23]. Thus, NDVI should be used to quantify the presence of chlorophyll using satellite images [24]. NDVI ranges between −1 to 1, with values below zero indicating either cloud or water, and values near zero indicating bare soil or sand. Mathematically, NDVI is notated as below:

$$NDVI = \frac{\rho_r - \rho_{NIR}}{\rho_r + \rho_{NIR}}$$

As with the previous subsection, this feature is computed in a function to calculate each image in the image collection. Specifically, GEE has the function of "ee.Image.normalizedDifference" to simplify the code required to calculate NDVI, as shown in Fig. 7.

Normalized difference water index (NDWI)

Specific coastal ecosystems, such as seagrass meadows, intertidal sand and subtidal sediment are often observed beneath the water surface by a satellite. Therefore, NDWI should be performed to ensure observation consistency. NDWI is used to highlight water features in satellite images [25] and ranges between −1 to 1, with values below zero indicating non-water bodies and near-zero values indicating shallow water. Mathematically, NDWI is notated as follows, and computed with the function of "ee.Image.normalizedDifference" (Fig. 8).

$$NDWI = \frac{\rho_{NIR} - \rho_r}{\rho_{NIR} + \rho_r}$$

Further steps

Performing both the image pre-processing and feature extraction steps detailed in this method paper will produce processed satellite images that will enable machine learning for the classification of coastal ecosystems. The specific steps to perform machine learning for this purpose are beyond the scope of this method paper. However, other studies exist that have described the various algorithms that can be used for coastal ecosystem mapping based on machine learning in GEE [26–28]. The methodological steps detailed in this study can also be adapted for purposes beyond ecosystem classification based on machine learning, for instance, for bathymetry [29], and seawater quality and coastal biodiversity monitoring [30,31]. The general methodological steps provided in this paper were designed to perform satellite image pre-processing and feature extraction directly in GEE that can be readily applied for further spatial analysis as required by practitioners, such as area managers, for various environmental management needs. The development of the readily adaptable workflow is essential to empower and enable local area managers to conduct monitoring of their area, particularly due to increasing stressors caused by anthropogenic activities and environmental changes that are impacting these ecosystems and the services they provide [32–34].

Method validation

To assess the practicality of this method, validation was performed by inviting practitioners to evaluate the steps described above. Invitations to participate in this validation step were sent to practitioners within the network of the authors of this paper, who were contacted individually to explain the purpose of this validation step. Out of five invitations sent, three practitioners agreed to participate and gave their written consents prior to the methods trial.

During the validation step, the practitioners were asked to follow the steps described above in the Method Details section, after which their feedback was collected. As part of the feedback, practitioners were asked to provide a Likert scale rating of 1 to 5 on the user-friendliness of the described methodology, with 5 being very easy to use. The participating practitioners were also asked to provide comments on the utility of the workflow. All of these are displayed in Table 1. Although the validation phase involved a limited number of participants, the validation results confirmed the accuracy and reliability of the proposed workflow. Nevertheless, we recognize that diverse practitioners may necessitate modifications to the pseudocodes presented here to accommodate their specific application requirements. Consequently, we have made the image preprocessing and feature extraction codes publicly available as an open-source resource [35], facilitating customization and adaptation for specialized research and operational needs.

Limitations

The methodological steps described here require basic understanding on the use of Python programming language, and thus may provide a barrier to its adoption by practitioners who are not familiar with Python as indicated by a comment provided by one of the participants during the validation process. However, open-access resources are available widely (e.g. [11–13]) to enable familiarization with the programme. In addition, the operation of GEE will require a working internet connection, hence limiting the applicability of our approach only to those having access to the internet.

```
# Assign a function to compute normalized difference vegetation index for
# each image on image collection
def NDVI(image):
    red = 'B4'
    nir = 'B8'
    ndvi = image.normalizedDifference([nir, red]).rename('ndvi')

    return image.addBands(ndvi, None, True)
```

Fig. 7. Normalized difference vegetation index.

```
# Assign a function to compute normalized difference water index for
# each image on image collection
def NDWI(image):
    green = 'B3'
    nir = 'B8'
    ndwi = image.normalizedDifference([green, nir]).rename('ndwi')

    return image.addBands(ndwi, None, True)
```

Fig. 8. Normalized difference water index.

Table 1
Participating practitioners' feedback.

Participant	Feedback
#1 (Practitioner from local government body).	• Rating of user-friendliness: 4. • Experience in using GEE: 1 year. • Comment: "*The methodology is clear enough.*".
#2 (Practitioner from non-governmental organization).	• Rating of user-friendliness: 3. • Experience in using GEE: None. • Comment: "*As a first-time user of GEE, it takes some time for me to get used to this methodology, especially as I am not familiar with Python. However, overall, the steps are relatively easy to follow.*".
#3 (Practitioner from local government body).	• Rating of user-friendliness: 5 • Experience in using GEE: 1 year, with relevant experience as GIS developer of 10 years. • Comment: "*Very easy to follow, and I did not encounter error.*".

Ethics statements

For the validation step, invited practitioners provided their written informed consent prior to their participation, ensuring their anonymity. Personal or demographical data were not collected from the participating practitioners.

Credit author statement

Ahmad Badruzzaman: Conceptualization, Methodology, Writing – Original Draft. **Prawesti Wulandari** and **Sainal Sainal**: Writing – Review and Editing. **Matthew Ashley**: Supervision, Writing – Review and Editing. **Susan Jobling** and **Melanie C. Austen**: Supervision, Writing – Review and Editing, Funding Acquisition. **Radisti A. Praptiwi**: Conceptualization, Project Administration, Supervision, Writing – Original Draft, Writing – Review and Editing, Funding Acquisition.

Declaration of interests

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

References

[1] C. Hattam, S. Broszeit, O. Langmead, R. Praptiwi, V. Lim, L. Creencia, H. Tran, C. Maharja, T. Setia, P. Wulandari, J. Sugardjito, J. Javier, E. Jose, L. Gajardo, A. Then, Y. Affendi, S. Johari, E. Justine, M. Hussein, H. Goh, M. Austen, A matrix approach to tropical marine ecosystem service assessments in South east Asia, Ecosyst. Serv. 51 (2021) 101346, https://doi.org/10.1016/j.ecoser.2021.101346.

[2] C. Maharja, R. Praptiwi, B. Roberts, K. Morrissey, M. White, N. Sari, F. Cholifatullah, J. Sugardjito, L. Fleming, Sea swimming and snorkeling in tropical coastal blue spaces and mental well-being: findings from Indonesian island communities during the COVID-19 pandemic, J. Outdoor Recreat. Tour. 41 (2023) 100584, https://doi.org/10.1016/j.jort.2022.100584.

[3] S. Geange, M. Townsend, D. Clark, J.I. Ellis, A.M. Lohrer, Communicating the value of marine conservation using an ecosystem service matrix approach, Ecosyst. Serv. 35 (2019) 150–163, https://doi.org/10.1016/j.ecoser.2018.12.004.

[4] A. Melet, P. Teatini, G. Le Cozannet, C. Jamet, A. Conversi, J. Benveniste, R. Almar, Earth observations for monitoring marine coastal hazards and their drivers, Surv. Geophys. 41 (2020) 1489–1534, https://doi.org/10.1007/s10712-020-09594-5.

[5] B.E. Mahrad, A. Newton, J.D. Icely, I. Kacimi, S. Abalansa, M. Snoussi, Contribution of remote sensing technologies to a holistic coastal and marine environmental management framework: a review, Remote Sens. (Basel) 12 (14) (2020) 2313, https://doi.org/10.3390/rs12142313.

[6] M.J. McCarthy, K.E. Colna, M.M. El-Mezayen, A.E. Laureano-Rosario, P. Méndez-Lázaro, D.B. Otis, G. Toro-Farmer, M. Vega-Rodriguez, F.E. Muller-Karger, Satellite remote sensing for coastal management: a review of successful applications, Env. Manage. 60 (2017) 323–339, https://doi.org/10.1007/s00267-017-0880-x.

[7] Q. Zhao, L. Yu, X. Li, D. Peng, Y. Zhang, P. Gong, Progress and trends in the application of Google Earth and Google Earth Engine, Remote Sens. (Basel) 13 (2021) 3778, https://doi.org/10.3390/rs13183778.

[8] M. Amani, A. Ghorbanian, S. Ahmadi, M. Kakooei, A. Moghimi, S. Mirmazloumi, S. Moghaddam, S. Mahdavi, M. Ghahremanloo, S. Parsian, Q. Wu, Google earth engine cloud computing platform for remote sensing big data applications: a comprehensive review, IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. 13 (2020) 5326–5350, https://doi.org/10.1109/JSTARS.2020.3021052.

[9] R. Prasai, T. Schwertner, K. Mainali, H. Mathewson, H. Kafley, S. Thapa, D. Adhikari, P. Medley, J. Drake, Application of Google Earth Engine python API and NAIP imagery for land use and land cover classification: a case study in Florida, USA, Ecol. Inf. 66 (2021) 101474, https://doi.org/10.1016/j.ecoinf.2021.101474.

[10] M. Fortnam, L. Evans, A. Amira Mas Ayu, L. Bastian, C.T.L. Creencia, H. Goh, B. Gonzales, K. Madarcos, C. Maharja, M. Mohd Iqbal, T. Ngoc Le, R. Praptiwi, J. Sugardjito, Q. Van Nguyen, W. Wan Nur Syazana, Polycentricity in practice: marine governance transitions in Southeast Asia, Env. Sci. Policy 137 (2022) 87–98, https://doi.org/10.1016/j.envsci.2022.08.010.

[11] Google for Developers, Google's Phyton class, Google, 2023 [Online]. Available: https://developers.google.com/edu/python [Accessed 28 6 2024].

[12] DataCamp, "Introduction to Phyton," DataCamp, [Online]. Available: https://www.datacamp.com/courses/intro-to-python-for-data-science?utm_source=google&utm_medium=paid_search&utm_campaignid=18600520486&utm_adgroupid=149047151470&utm_device=c&utm_keyword=python%20free%20course&utm_matchtype=e&utm_network=g&utm_adpostion=&u. [Accessed 28 6 2024].

[13] Microsoft, "Phyton for beginners," Microsoft, [Online]. Available: https://learn.microsoft.com/en-us/training/paths/beginner-python/. [Accessed 28 6 2024].

[14] C.P. Dave, R. Joshi, S. Srivastava, A survey on geometric correction of satellite imagery, Int. J. Comput. Appl. 116 (12) (2015) 24–27.

[15] D.R. Lyzenga, N.P. Malinas, F.J. Tanis, Multispectral bathymetry using a simple physically based algorithm, IEEE Trans. Geosci. Remote Sens. 44 (8) (2006) 2251–2259, https://doi.org/10.1109/TGRS.2006.872909.

[16] S. Kay, J. Hedley, S. Lavender, Sun glint correction of high and low spatial resolution images of aquatic scenes: a review of methods for visible and near-infrared wavelengths, Remote Sens. (Basel) 1 (4) (2009) 697–730, https://doi.org/10.3390/rs1040697.

[17] P. Wicaksono, P. Aryaguna, W. Lazuardi, Benthic habitat mapping model and cross validation using machine-learning classification algorithms, Remote Sens. (Basel) 11 (11) (2019) 1279, https://doi.org/10.3390/rs11111279.

[18] R. Martínez Prentice, M. Villoslada Peciña, R. Ward, T. Bergamo, C. Joyce, K. Sepp, Machine learning classification and accuracy assessment from high-resolution images of coastal wetlands, Remote Sens. (Basel) 13 (18) (2021) 3669, https://doi.org/10.3390/rs13183669.

[19] D. Lyzenga, Passive remote sensing techniques for mapping water depth and bottom features, Appl. Opt. 17 (3) (1978) 379–383, https://doi.org/10.1364/AO.17.000379.

[20] P. Wicaksono, S. Wulandari, W. Lazuardi, M. Munir, Sentinel-2 images deliver possibilities for accurate and consistent multi-temporal benthic habitat maps in optically shallow water, Remote Sens. Appl.: Soc. Environ. 23 (2021) 100572, https://doi.org/10.1016/j.rsase.2021.100572.

[21] J. Uitz, H. Claustre, A. Morel, S. Hooker, Vertical distribution of phytoplankton communities in open ocean: an assessment based on surface chlorophyll, J. Geophys. Res.: Oceans 11 (C8) (2006), https://doi.org/10.1029/2005JC003207.

[22] Y. Zhao, C. Yan, S. Lu, P. Wang, G. Qiu, R. Li, Estimation of chlorophyll content in intertidal mangrove leaves with different thicknesses using hyperspectral data, Ecol. Indic. 106 (2019) 105511, https://doi.org/10.1016/j.ecolind.2019.105511.

[23] J. Silva, Y. Sharon, R. Santos, S. Beer, Measuring seagrass photosynthesis: methods and applications, Aquat. Biol. 7 (1–2) (2009) 127–141, https://doi.org/10.3354/ab00173.

[24] J.W. Rouse, R.H. Haas, J.A. Schell, D.W. Deering, Monitoring vegetation systems in the great plains with ERTS (Earth Resources Technology Satellite), NASA Spec. Publ. 351 (1) (1974).

[25] S. McFeeters, The use of the normalized difference water index (NDWI) in the delineation of open water features, Int. J. Remote Sens. 17 (7) (1996) 1425–1432.

[26] M. Bessinger, M. Lück-Vogel, A. Skowno, F. Conrad, Landsat-8 based coastal ecosystem mapping in South Africa using random forest classification in Google Earth Engine, S. Afr. J. Bot. 150 (2022) 928–939, https://doi.org/10.1016/j.sajb.2022.08.014.

[27] A. horbanian, S. Zaghian, R. Asiyabi, M. Amani, A. Mohammadzadeh, S. Jamali, Mangrove ecosystem mapping using Sentinel-1 and Sentinel-2 satellite images and random forest algorithm in Google Earth Engine, Remote Sens. (Basel) 13 (13) (2021) 2565, https://doi.org/10.3390/rs13132565.

[28] A. Ahmed, F. Mutua, B. Kenduiywo, Monitoring benthic habitats using Lyzenga model features from Landsat multi-temporal images in Google Earth Engine, Model. Earth Syst. Environ. 7 (3) (2021) 2137–2143, https://doi.org/10.1007/s40808-020-00960-1.

[29] D. Traganos, D. Poursanidis, B. Aggarwal, N. Chrysoulakis, P. Reinartz, Estimating satellite-derived bathymetry (SDB) with the google earth engine and sentinel-2, Remote Sens. (Basel) 10 (6) (2018) 859, https://doi.org/10.3390/rs10060859.

[30] I. Kwong, F. Wong, T. Fung, Automatic mapping and monitoring of marine water quality parameters in Hong Kong using Sentinel-2 image time-series and Google Earth Engine cloud computing, Front. Mar. Sci. 9 (2022) 871470, https://doi.org/10.3389/fmars.2022.871470.

[31] M. León-Pérez, A. Reisinger, J. Gibeaut, Spatial-temporal dynamics of decaying stages of pelagic sargassum spp. Along shorelines in Puerto Rico using Google Earth Engine, Mar. Pollut. Bull. 188 (2023) 114715, https://doi.org/10.1016/j.marpolbul.2023.114715.

[32] N.J. Beaumont, M. Aanesen, M.C. Austen, T. Börger, J.R. Clark, M. Cole, T. Hooper, P.K. Lindeque, C. Pascoe, K.J. Wyles, Global ecological, social and economic impacts of marine plastic, Mar. Pollut. Bull. 142 (2019) 189–195, https://doi.org/10.1016/j.marpolbul.2019.03.022.

[33] C. Maharja, R. Praptiwi, Y. Purwanto, Understanding the cultural impacts of climate change harms on small-scale fisher communities through the lens of cultural ecosystem services, Marit. Stud. 22 (4) (2023) 41, https://doi.org/10.1007/s40152-023-00332-2.

[34] C. Maharja, R.A. Praptiwi, S. Sainal, P. Wulandari, M. Ashley, K.J. Wyles, J. Roy, I.G. Hendrawan, S. Jobling, M.C. Austen, Multiple negative impacts of marine plastic pollution on tropical coastal ecosystem, Ocean. Coast. Manage. 258 (2024) 107423, https://doi.org/10.1016/j.ocecoaman.2024.107423.

[35] PISCES Partnership, "Open source codes satellite image preprocessing and feature extraction" [Online], Available: https://github.com/piscesesaunggul/Open-source-codes-satellite-image-preprocessing-and-feature-extraction.git.