Revista Brasileira de Cartografia
ISSN 1808-0936 | https://doi.org/10.14393/revbrascartogr
Sociedade Brasileira de Cartografia, Geodésia, Fotogrametria e Sensoriamento Remoto

Integrating Open Data Cube and Brazil Data Cube Platforms for Land Use and Cover Classifications

Integração das Plataformas Open Data Cube e Brazil Data Cube para Classificação de Uso e Cobertura da Terra

Felipe Menino Carlos [1], Vitor Conrado Faria Gomes[2], Gilberto Ribeiro de Queiroz [1], Felipe Carvalho de Souza [1], Karine Reis Ferreira[1] and Rafael Santos [1]

1 National Institute for Space Research (INPE), Earth Observation and Geoinformatics Division, São José dos Campos – SP, Brazil. efelipecarlos@gmail.com, gilberto.queiroz@inpe.br, felipe.carvalho@inpe.br, karine.ferreira@inpe.br, rafael.santos@inpe.br
ORCID: https://orcid.org/0000-0002-3334-4315
ORCID: https://orcid.org/0000-0001-7534-0219
ORCID: https://orcid.org/0000-0002-5826-1700
ORCID: https://orcid.org/0000-0003-2656-5504
ORCID: https://orcid.org/0000-0002-8313-6688
2 Institute for Advanced Studies (IEAv), C4ISR Division, São José dos Campos – SP, Brazil. vitorvcfg@fab.mil.br.
ORCID: https://orcid.org/0000-0003-3239-2160

Recebido: 04.2021 | Aceito: 06.2021

Abstract: The potential to perform spatiotemporal analysis of the Earth's surface, fostered by a large amount of Earth Observation (EO) open data provided by space agencies, brings new perspectives to create innovative applications. Nevertheless, these big datasets pose some challenges regarding storage and analytical processing capabilities. The organization of these datasets as multidimensional data cubes represents the state-of-the-art in analysis-ready data regarding information extraction. EO data cubes can be defined as a set of time-series images associated with spatially aligned pixels along the temporal dimension. Some key technologies have been developed to take advantage of the data cube power. The Open Data Cube (ODC) framework and the Brazil Data Cube (BDC) platform provide capabilities to access and analyze EO data cubes. This paper introduces two new tools to facilitate the creation of land use and land over (LULC) maps using EO data cubes and Machine Learning techniques, and both built on top of ODC and BDC technologies. The first tool is a module that extends the ODC framework capabilities to lower the barriers to use Machine Learning (ML) algorithms with EO data. The second tool relies on integrating the R package named Satellite Image Time Series (`sits`) with ODC to enable the use of the data managed by the framework. Finally, water mask classification and LULC mapping applications are presented to demonstrate the processing capabilities of the tools.

Keywords: Earth observation data cube. Land use and land cover classification. Open data cube. Brazil data cube.

Resumo: O potencial para realizar análises espaço-temporais da superfície terrestre, fomentado por uma grande quantidade de dados abertos de Observação da Terra (EO) fornecidos por agências espaciais, traz novas perspectivas para a criação de aplicações inovadoras. No entanto, estes grandes conjuntos de dados trazem novos desafios em relação às capacidades de armazenamento e processamento analítico. A organização desses conjuntos como cubos de dados representa o estado da arte na disponibilização de dados prontos para análise e, consequentemente, para realização de extração de informação. Os cubos de dados de EO podem ser definidos como um conjunto de séries temporais de imagens alinhadas no tempo e espaço. Algumas tecnologias-chave foram desenvolvidas para aproveitar o poder dos cubos de dados. O framework Open Data Cube (ODC) e a plataforma Brazil Data Cube (BDC) fornecem recursos para acessar e analisar esses dados. Este trabalho introduz duas novas ferramentas para facilitar a criação de mapas uso e cobertura da terra (LULC), utilizando cubos de dados e técnicas de Aprendizado de Máquina (ML), criadas com base nas tecnologias ODC e BDC. A primeira ferramenta é um módulo que estende as capacidades do ODC para facilitar a utilização de algoritmos de ML com dados de EO. A segunda ferramenta realiza a integração do pacote R denominado Satellite Image Time Series (`sits`) com o ODC para possibilitar o uso dos dados gerenciados por este framework. Para demonstrar o potencial de processamento das ferramentas desenvolvidas duas aplicações foram geradas. A primeira trata da criação de uma máscara de água, enquanto a segunda da produção de um mapa de LULC.

Palavras-chave: Cubo de dados de observação da terra. Classificação de uso e cobertura da terra. Open data cube. Brazil data cube.

1 INTRODUCTION

In recent years, the amount of Earth Observation (EO) data available has grown, driven by technological advances in acquisition and storage equipment and space agency policies that make their data repositories freely available. Soille et al. (2018) estimate that the annual data volume produced by well-known EO platforms, including Landsat 7 and 8, Terra and Aqua, and the Sentinel 1, 2, and 3, can reach 04 petabytes. The EO scientific community has developed and made available many specialized tools to address the challenge of managing, processing, and analyzing these large volumes of data. These tools are often provided as software infrastructures that manage large time-series of EO data using multidimensional array concepts, making it easy for users to access and use Analysis-Ready Data (ARD). The paradigm adopted by these systems is called Earth Observation Data Cubes (EODC) (GIULIANI et al., 2019). Despite using the same approach, different solutions and technologies are often adopted to build these systems, resulting in lack of interoperability (NATIVI et al., 2017).

The ODC (ODC, 2021) is considered one of the leading frameworks using the freely available EODC concept (GOMES et al., 2020). In 2019, 56 initiatives in the world were deploying and evaluating the use of the ODC in an institutional context (KILLOUGH, 2018). This system is composed of tools and services made available in modules that can be combined as needed by users. Recently, ODC has been used by different projects to manage EODC for a specific country, such as the Australian Data Cube (LEWIS et al., 2017), Swiss Data Cube (GIULIANI et al., 2017), and Africa Regional Data Cube[1] (KILLOUGH, 2018).

The Brazil Data Cube (BDC) project is an initiative from the Brazilian National Institute for Space Research (INPE) committed to creating EODC for the whole country based on medium resolution satellite images, including Sentinel-2/MSI, Landsat-8/OLI, and CBERS-4/AWFI. A relevant application built up from this data is land use and land cover (LULC) maps created with time-series analysis methods and machine learning (FERREIRA et al., 2020). LULC maps can be used to assess changes on the Earth's surface caused by anthropic activities and thus establish environmental and policy control. For instance, Simoes et al. (2020) created LULC maps from 2001 to 2017 for Mato Grosso State/Brazil based on a time-series analysis, providing information about agriculture and natural vegetation dynamics over time. Sanchez et al. (2019) used time-series from EODCs and Deep Learning techniques to identify deforested areas. The BDC project is currently developing a computational platform using big data technologies and cloud computing environments to support this type of application.

Carlos et al. (2021) presents the integration of the ODC framework and the BDC platform, which allowed ODC tools to use data products available in the BDC platform. Although ODC provides several tools to manage EODCs, it does not have ready-to-use functionalities for LULC mapping. Therefore, creating LULC maps from the raw ODC API is a hard task as it requires advanced programming skills. On the other side, the BDC platform makes available an R package named Satellite Image Time Series (`sits`) designed to support LULC mapping through ML algorithms and EODCs.

Thus, considering these characteristics of each technology integrated by Carlos et al. (2021) and the importance of LULC classification, we introduce two new tools in this work. The first one, called `datacube-classification,` extends the ODC functionalities and makes it simpler to use ML algorithms and EODC for LULC mapping. The second, `odc-sits`, is a tool written in the R programming language that adds some functionalities to `sits` that enable it to access EODCs managed by ODC. Both tools help improve the integration between ODC and BDC, which benefit users from such technologies with new functionalities and open new EODC usage perspectives.

The remainder of this paper is organized as follows. In Section 2, we highlight the main components of both systems involved in the integration presented by Carlos et al. (2020) and the strategy behind it. The `datacube-classification` library built on top of the ODC is introduced in Section 3. Section 4 shows the `odc-sits` package to access ODC data in R language. Section 5 unveils the power of such integration, besides the new tools developed, through two applications: one that produces a water mask and another that performs the classification of LULC information from the Brazilian EO data cube. Finally, in Section 6, we make the

[1] Since 2020, this has been renamed as Digital Earth Africa (DE Africa).

final considerations regarding the challenges encountered and the next steps that will be developed.

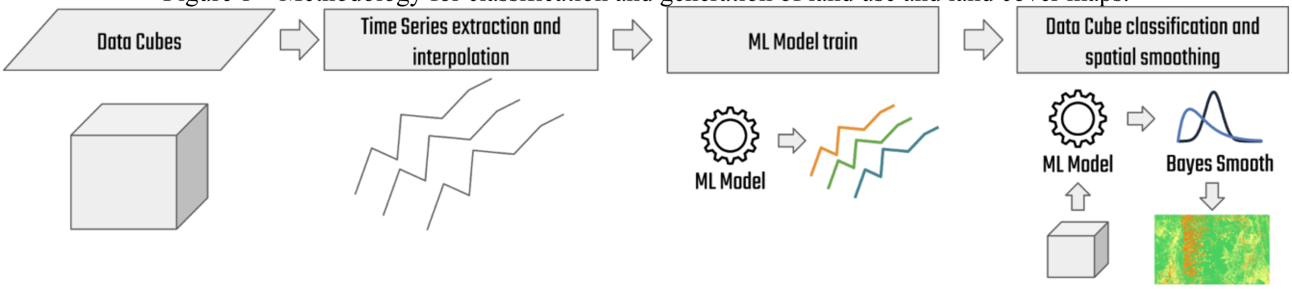
2 MATERIALS AND METHODS

2.1 Brazil Data Cube

The BDC project was created with the following objectives (FERREIRA et al., 2020): (i) Produce ARD and EODC from medium-resolution satellite images for Brazil; (ii) Creation of new methods and techniques for storing, processing, and analyzing large volumes of EO data; and (iii) Generation of LULC maps using satellite image time-series extracted from EODC data and ML techniques. The BDC project carries out the creation of data and software products to meet these objectives. The software tools include a SpatioTemporal Asset Catalog (STAC) to search and retrieve EODC data besides the `sits` analytical package (CAMARA et al., 2018) to generate LULC maps. These features enable users to access and process effectively the data products produced by BDC using data from the CBERS-4/AWFI, Landsat-8/OLI, and Sentinel-2/MSI sensors.

In the BDC project, the generation of LULC classification maps is performed by analyzing time-series extracted from EODCs (FERREIRA et al., 2020). This process is illustrated in Figure 1. In the first step, the time-series associated with the location of labeled samples are extracted from the data cubes. Pixel values related to cloud cover are replaced through a time-series interpolation method. Then, the ML model is trained using these time-series associated with the sample labels. Finally, the trained model is applied to the data cube to produce the LULC map. Post-processing is optionally performed using spatial smoothing approaches to reduce noise and then improve the classification result.

Figure 1 – Methodology for classification and generation of land use and land cover maps.


Data Cubes
Time Series extraction and interpolation
ML Model train
Data Cube classification and spatial smoothing
ML Model
ML Model
Bayes Smooth

Source: Adapted from Ferreira et al. (2020).

Currently, the methodology described is applied in the BDC platform through the analytical package `sits`, an open-source R package created for performing analysis, visualization, and classification of satellite image time-series data. The package was designed to support the complete LULC classification workflow with big satellite image datasets. Therefore, `sits` automatically optimizes the data load using chunking methods based on parallel processing. The package also includes sample selection, time-series clustering, ML model training, and classification evaluation using spatial-ready functions. For classification, `sits` provides ML algorithms such as Multilayer Perceptron Neural Network, Random Forest, and Support Vector Machines.

2.2 Open Data Cube

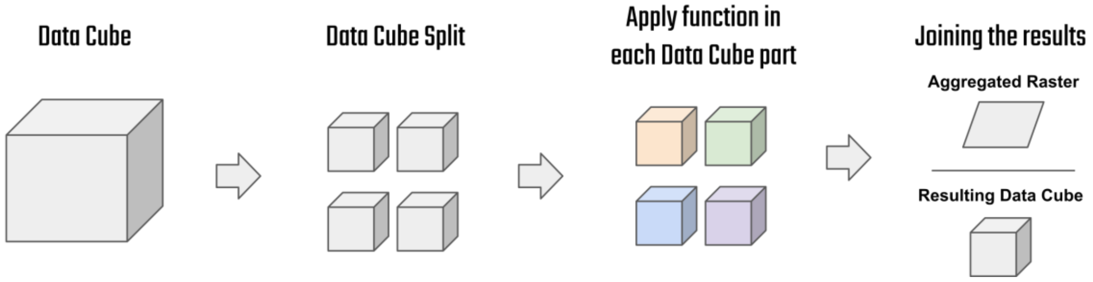
The Open Data Cube (ODC) is a modular framework that provides data structures and tools for handling large volumes of Earth observation data. It can catalog EO datasets and manage them through command-line tools (CLI), Python libraries, and web services. The users can enable and combine ODC modules according to their needs.

The main ODC module is the `datacube-core`, responsible for indexing, searching, and loading data. This module stores catalog metadata into a PostgreSQL database and records spatio-temporal information of datasets. The other modules of the ODC ecosystem use the `datacube-core` to search for metadata and then access the underlying image files. The `datacube-explorer` module allows data to be searched and displayed in a web map application. The `datacube-ows` offer an OGC web services interface. The `datacube-stats` can run aggregate functions over large spatio-temporal datasets.

In this work, the `datacube-stats` module was extended to process EODC using ML techniques. This module provides a high-level CLI that allows users to describe the statistical processing through a configuration file in YAML format. In this file, the user can inform the input data involved in the processing by providing the collection name. It can also define the area of interest with the spatial and temporal extension. In addition, the user can specify the function applied in the data, which allows the usage of predefined functions or user-defined functions. In the latter case, it is necessary to inform how to import the Python module that contains such a function. Finally, the user must specify the output format for the processed data and the file name pattern of these files.

This feature allows researchers to extract information from the EODCs by performing this operation without implementing code or directly using parallel processing tools. In the configuration file, the researcher can inform how he wants data to be fragmented by specifying the amount and size of data chunks. The outputs of the `datacube-stats` operations can represent aggregated data over time or even a new data cube, depending on how the operations are described. Figure 2 summarize the workflow of `datacube-stats`.

Figure 2 – datacube-stats operation.



Source: The authors (2021).

Out of the box, `datacube-stats` provides statistical operations, such as mean, median, and geometric median, which allow the temporal aggregation of the data. The module also makes a class in Python available as an extension hotspot to add new functions to this tool. This work uses this feature to extend this package to generate classification maps using ML algorithms used in the BDC.

2.3 STAC2ODC: Integrating the BDC data products through ODC instances

The `STAC2ODC` (STAC to ODC) tool introduced by Carlos et al. (2020) was designed to allow access to the BDC data products through the tools and services available in ODC. This tool has a CLI that reads metadata from image and data cube collections and their items, available in the `bdc-stac` web service, and indexes this information in the catalog managed by an ODC instance. Therefore, it maps the BDC data model to the ODC data model. The authors mentioned above used this tool to automatically register all BDC data products in an ODC instance running on the BDC infrastructure (`bdc-odc`).

Carlos et al. (2020) also integrated the ODC modules `datacube-explorer` and `datacube-ows` with BDC data products. These tools enable the dissemination and visualization of the BDC's data products. The `datacube-explorer` provides a web interface to perform spatio-temporal searches of the data and identify products available for use. The `datacube-ows` bring to the `bdc-odc` instance the implementation of OGC standards such as WMS and WCS that can be used to access the available data in the ODC catalog.

2.4 Extending ODC and BDC

The technologies covered up to this section realize the integration such that users of BDC data products can use all the tools provided by ODC. Nevertheless, this current work aims to allow users of the BDC platform to use data served by ODC instances directly through the R language. Thus, users of technologies developed in the BDC scope will have a seamless integration using data from other data cubes based on ODC.

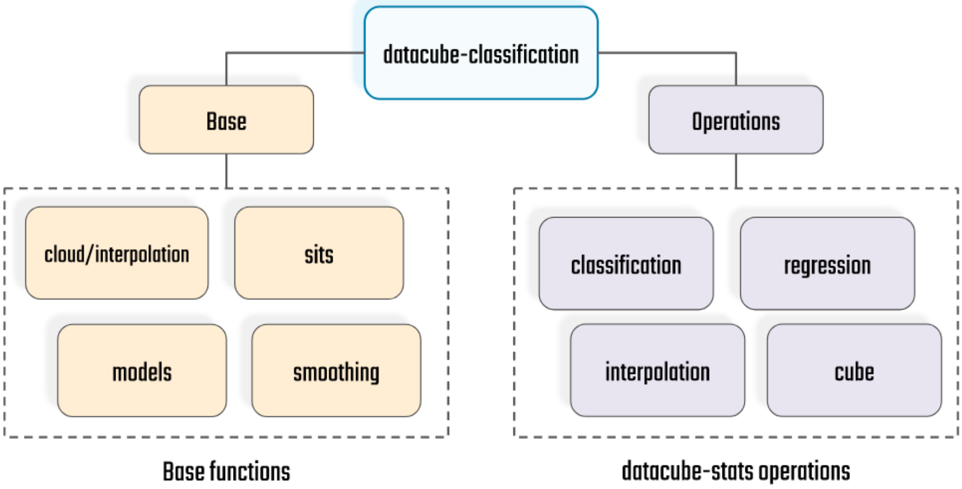
Furthermore, it is also intended to provide to ODC users an easiest way to use ML algorithms, especially for creating LULC maps. Hence, it is necessary to augment the ODC ecosystem with a module that hides all the complexity of dealing with ML algorithms, EO data, and time-series extraction. It is also desired that such a module allows for rapid integration of well-known ML packages in Python.

In conclusion, such functionalities would allow the R programming user community in the BDC project to take advantage of data managed by ODC instances. On the other hand, the Python users from ODC would have an alternative solution to perform ML analysis through the ODC ecosystem. To accomplish both goals, two applications were developed. The first, called `datacube-classification`, extends the `datacube-stats` module to allow Python users to easily extract time-series, train ML models, and classify data from EODC. The second tool, named `odc-sits,` extends `sits` functionalities to enable it to access EODCs managed by ODC. The details of the creation of these tools are presented in the following sections.

3 THE DATACUBE-CLASSIFICATION LIBRARY

The `datacube-classification` library was implemented following the concept of modules and submodules through the extension of `datacube-stats`. This module was chosen because it allows users and researchers to easily describe computational flows with a high level of abstraction and low complexity to process large volumes of data. As presented in Figure 3, the library has two main modules, `Base` and `Operations`. These are responsible, respectively, for providing the basic functions and operations that can be used in `datacube-stats`.

Figure 3 – datacube-classification modules and submodules.



Source: The authors (2021).

Each of the modules of `datacube-classification` was structured in submodules facilitating new functionalities and future maintenance of the code. Below, these modules and submodules are detailed.

3.1 Base module

The `Base` module contains submodules to provide necessary functionalities to compose the `Operations` module. The `sits` submodule implements functionalities for the extraction and visualization of time-series from EODCs. For extraction, it is possible to define specific locations, or use, in an optimized way, all the time-series represented in a data cube.

The `cloud/interpolation` submodule provides functionality that allows the removal of cloud influence on the pixels extracted from the data cube. This influence modifies the values and significantly altering the spectro-temporal behavior represented by the extracted series. Note that the cloud identification is performed based on the cloud mask provided by the user for each image in the EODC. The removed pixel is filling by applying temporal linear interpolation. All operations in the sits submodule are already integrated with this functionality. Thus, if the user specifies the cloud mask in the extraction operation, the interpolation process described above is performed automatically.

The models submodule facilitates the use of ML algorithms, providing functionality for training and applying them in EODC. The features provided can be used considering classification and regression algorithms. Finally, to improve the classification results, the smooth module provides functionalities to allow spatial smoothing. The smoothing functions used were implemented originally in sits package. To import these functions, the Carma[2] library was used.

3.2 Operations module

All features implemented in Base are written in generic form and can be imported and consumed in any Python project. The functions operate on N-Dimensional data structures provided by the xarray library (HOYER; HAMMAN, 2017). The Operations module uses these features to create operations that can be applied to the data through datacube-stats. These operations are implemented using the extension hotspot mentioned earlier.

The classification submodule allows the application of an ML model to perform the EODC classification. The application of this submodule requires that the model is already trained and serialized in a joblib[3] file. The user can do model training with the functionality provided by the Base module in a Python script external to this process. Furthermore, the model specified by the user must be implemented following the interface and algorithm definitions of the scikit-learn library (PEDREGOSA et al., 2011). Any algorithm that implements these interfaces, including all algorithms in the scikit-learn itself and other implementations, such as the Random Forest implementation in the skranger[4] library, are able to be run with the datacube-classification library.

In the regression submodule, was implemented a function that allows the use of the Linear Spectral Mixture Model (MLME) in each time scene from an EODC to generate fractions images that can be used as auxiliary data to the classification process. This functionality uses the Multiple Endmember Spectral Mixture Model (MESMA) implementation provided by the RStoolbox R package (LEUTNER et al., 2017). The usage of this function in the Python environment was done with the rpy2[5] library. As specified by Shimabukuro and Ponzoni (2019), the use of MLME depends on the definition of endmembers, representing the behavior of the elements being modeled. To use this operation, the endmembers used for all scenes must be provided by the user.

Two auxiliary operations submodules have also been implemented in the datacube-classification library. First, the interpolation submodule allows the interpolation of all time-series in an EODC, based on the cloud mask provided as input by the user. Thus, it enables the generation of already interpolated data cubes to be made available to users as new indexed collections in the ODC without cloud influence. The cube submodule allows new data cubes to be generated based on user-defined functions. These functions can then manipulate the input bands to compose cubes with new attributes. For example, it is possible to create new EODCs with spectral indices, which can be helpful in different activities.

4 THE ODC-SITS PACKAGE

The odc-sits package was created to allow the features available in the sits package for R language to be used with indexed data in an ODC instance. As illustrated in Figure 4, the package provides two main functionalities to enable integration between the sits and ODC environment. The first functionality, implemented through the odc_search function, allows querying the ODC metadata database using spatio-

[2] Carma documentation: https://carma.readthedocs.io/

[3] joblib persistance: https://scikit-learn.org/stable/modules/model_persistence.html

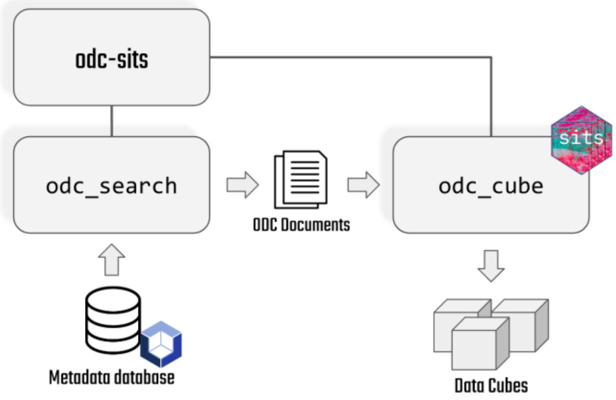
[4] skranger code repository: https://github.com/crflynn/skranger

[5] rpy2 documentation: https://rpy2.github.io/

temporal information. It returns ODC documents with reference to the data that meet the query criteria. With this function, only the metadata of the searched data is returned, making this operation fast. Furthermore, this metadata is represented by a list data structure that can be used for more specific filters, such as selecting data based on the available spectral bands.

After performing the search, the `odc_cube` function can load the data represented by each of these metadata. It reads the metadata and extracts the information from the stored files. This information is then organized in the format required by the `sits` package. Then the data cubes are materialized in the R environment using the data structures offered by the `sits` package. In the `odc_cube`, users can also specify how the library must treat the specific organization of the data. For example, the user can specify a rule that considers the division of the data cube into tiles. For this, the user can specify how the files will be loaded using their naming convention. Thus, the organizational structures present in the data can be maintained after load.

Figure 4 – odc-sits usage process.



Source: The authors (2021).

Currently, the `odc-sits` package requires the spatio-temporal alignment of the data. All the EODC made available by the BDC project in the `bdc-odc` instance is already aligned. This requirement can also be guaranteed through data ingestion, an operation available in the `datacube-core`.

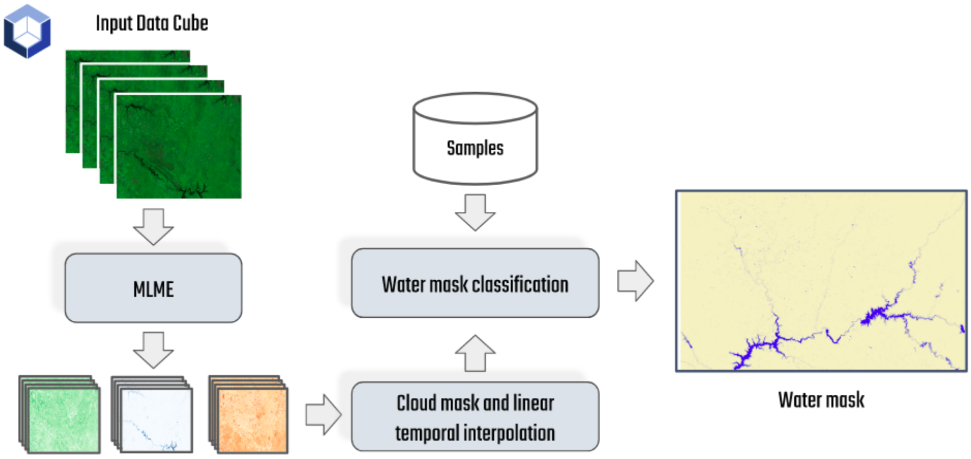
5 APPLICATIONS

To demonstrate the possibilities of using the technologies developed in this work were created two examples of applications. The first example is the generation of water masks through classification of fraction data cubes generated by MLME. In the second example, is performed a LULC mapping. Both examples were performed using the ODC instance of the BDC project and the tools previously presented.

5.1 Water mask using fraction data cubes

The water mask generation is made using the `datacube-classification` library, applied through the `datacube-stats` tool. Figure 5 illustrates the methodology used in this example. Initially, the fraction image data cube is generated by applying MLME to each EODC input scene. After that, the fraction image cube has the cloud influence removed. Where data is missing due to the presence of clouds, temporal linear interpolation is applied. Finally, the supervised ML algorithm is trained and applied. For this, a training dataset is used, which has samples associated with specific temporal behavior. This behavior refers to each fraction modeled with the MLME and is extracted from the data cube. With this information, the model is trained and then used to classify the entire cube of fraction image data.

Figure 5 – Methodology for water mask generation.


Input Data Cube
Samples
MLME
Water mask classification
Cloud mask and linear temporal interpolation
Water mask

Source: The authors (2021).

For this example, the Landsat-8/OLI data cube was used, with a temporal composition of 16 days. Scenes from the data cube were used for the Western Minas Gerais State/Brazil region (Figure 6A) from 01/2018 to 12/2018. To classify the input data, a Random Forest algorithm with 1000 trees was used. The algorithm was trained with 200 samples (Figure 6B), divided equally between Water and Non-Water classes. Listing 1 shows the basic structure of the YAML configuration file used in this application. It is important to highlight the `statistic_args` section. The value of `impl` (`datacube_classification.operations.classification.ScikitLearnClassifier`) indicates to the **`datacube-stats`** a function that should be applied to the data. In this case, is used the classification function provided by `datacube-classification`, which uses the Random Forest implementation provided by the `skranger` library. This classification function receives as input the data and the model that needs to be used. The model is specified through the `classification_model` value, a joblib-serialized file with an already trained model. The complete configuration file is available on the `datacube-classification` GitHub repository[6].

Listing 1 – datacube-stats YAML configuration file for water mask processing using datacube-classification extension.

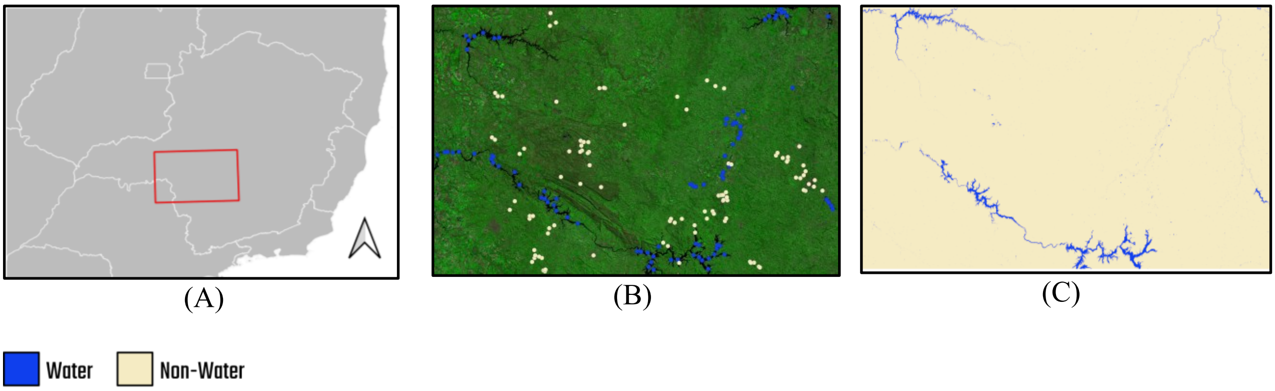
```
sources:
  - product: LC8_30_16D_STK_1_MixtureModel
    measurements: [ GROUND_FRACTION, VEGETATION_FRACTION, WATER_FRACTION ]
    time: [ 2018-01-01, 2020-12-25 ]
# (omitted code)
output_products:
 - name: LC8_30_16D_STK_1_MixtureModel_classification
   statistic: external
   statistic_args:
     impl: datacube_classification.operations.classification.ScikitLearnClassifier
     classification_model: "rfor1000.joblib"
```

Source: The authors (2021).

This application has used the global endmembers generated by Souza and Small (2017). These endmembers allow the MLME to be applied to create an EODC with the water, shade, and vegetation fractions. The results of applying this approach, using the implemented tool, are presented in Figure 6C.

[6] datacube-classification source code repository: https://github.com/brazil-data-cube/datacube-classification

Figure 6 – (A) Study area used for the generation of the water mask; (B) Distribution of the sample set used; (C) Water mask generated with the application of the described methodology.



Source: The authors (2021).

5.2 Land Use and Land Cover Mapping

The classification performed in this example was based on the sits features, made available with the integration performed by odc-sits. Once the EODC scenes are filtered and loaded, they are used to train a Random Forest model. Then, the trained model is used to generate the LULC map by classifying time-series from the data cube.

For the classification, was used the CBERS-4/AWFI EODC for the Western Bahia State/Brazil region (Figure 7A). To train the Random Forest has used a dataset with 922 LULC samples (Figure 7B), divided into Pasture, Agriculture, and Natural Vegetation classes. Each sample was associated with a time-series containing the following bands and vegetation indices attributes: Red (Band 15), Green (Band 14), Blue (Band 13), NIR (Band 16), Enhanced Vegetation Index (EVI), and Normalized Difference Vegetation Index (NDVI).

Listing 2 shows a code snippet to illustrate the usage of odc-sits to search and load data from an ODC instance and to classify this dataset using sits algorithms. The script source code is available on the odc-sits GitHub repository[7]. The LULC map generated by applying the ML model is shown in Figure 7C.

Listing 2 – Example of R source-code used for LULC classification using odc-sits.

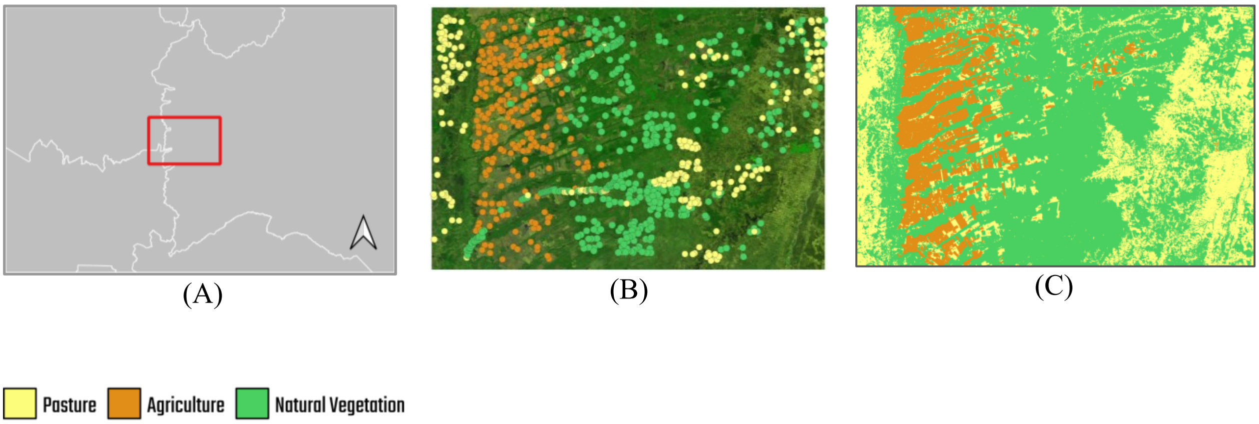
```
datasets <- odc_search(
  collection = "CB4_64_16D_STK_1",
  start_date = "2018-09-01",
  end_date   = "2019-08-01")
cube  <- odc_cube(index, "CBERS-4", "AWFI", datasets)

rfor  <- sits_train(samples, ml_method = sits_rfor(num_trees = 1000))
probs <- sits_classify(
              data    = cube, ml_model   = rfor,
              memsize = 16   , multicores = 8)
```

Source: The authors (2021).

[7] odc-sits source code repository: https://github.com/brazil-data-cube/odc-sits

Figure 7 – (A) Study area used for generating the LULC map; (B) Distribution of the sample set used; (C) LULC map generated with the classification process.



Source: The authors (2021).

6 FINAL REMARKS

The previous work (CARLOS et al., 2020), focused on allowing ODC functionalities to be used with BDC data products. Here, the integration between the Brazil Data Cube and Open Data Cube technologies was expanded with two new tools: `datacube-classification` and `odc-sits`. The first, named `datacube-classification`, brings to the ODC ecosystem the possibility to perform all the main methodological steps involved in the EODC data classification as applied in the BDC project. Its extensible design allows it to implement many different workflows for generating LULC maps. The second tool, named `odc-sits`, introduces an R package that allows EO data products available in an ODC instance to be used with the `sits` package developed and maintained by the Brazil Data Cube project.

To support the claim of this article, the developed tools were validated using two applications that often make up the analytical routines of EO data users. In the first of these, the Linear Spectral Mixture Model was applied to the input dataset to generate an EODC of fraction images used in the classification process. These fraction images were used for training the ML algorithm and identifying water bodies. With the second application, LULC maps were produced using the ML algorithms available in the `sits` package and the data available in the `bdc-odc` instance.

These results show that the tools presented in this paper have great potential for applications that make joint use of ML algorithms and EODC. Furthermore, it is expected that the contribution of this paper can be useful to users of the BDC platform and other members of the EO community who wish to apply classification methodologies to data managed in ODC instances.

6.1 Code and data availability

All the tools developed in this work are available in the Brazil Data Cube organization on GitHub[8]. The samples used were provided by Ferreira et al. (2020) and are publicly available in this data repository[9] of the Brazil Data Cube project.

Acknowledgments

The authors thank the Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) (Process n° 130727/2020-2) and the Brazil Data Cube subproject of the Project Monitoramento Ambiental dos Biomas Brasileiros, financed with resources from the Fundo Amazônia, through the financial collaboration

[8] Brazil Data Cube organization on GitHub: https://github.com/brazil-data-cube

[9] Training samples: https://brazildatacube.dpi.inpe.br/public/bdc-article/training-samples/

BNDES and FUNCATE nº 17.2.0536.1

Author Contributions

Conceptualization, F.M.C.,V.C.F.G., G.R.Q., K.R.F.; Data curation: F.M.C., F.C.S.; Formal Analysis, F.M.C., V.C.F.G., F.C.S.; Funding acquisition: G.R.Q., K.R.F.; Investigation, F.M.C., F.C.S; Methodology, F.M.C.,V.C.F.G., G.R.Q.,R.S; Project administration, G.R.Q., K.R.F., R.S.; Resources, G.R.Q., K.R.F.; Software, F.M.C., F.C.S; Supervision, G.R.Q., K.R.F., R.S.; Validation, F.M.C., V.C.F.G., F.C.S; Visualization, F.M.C., V.C.F.G.; Writing—Original Draft F.M.C.,V.C.F.G.,G.R.Q.; Writing—Review and editing, F.M.C., V.C.F.G., G.R.Q.

Conflicts of Interest

The authors declare no conflict of interest.

References

CAMARA, G.; SIMOES, R.; ANDRADE, P.R.; MAUS, V.; SÁNCHEZ, A.; DE ASSIS, L.F.F.G.; SANTOS, L.A. YWATA, A.C.; MACIEL, A.M.; VINHAS, L.; et al. e-sensing/sits: Version 1.12.5; **Zenodo**: Geneva, Switzerland, 2018.

CARLOS, F. M.; GOMES, V. C. F.; QUEIROZ, G. R. De.; FERREIRA, K. R.; SANTOS, R. Integração dos ambientes Brazil Data Cube e Open Data Cube. **Proceedings XXI GEOINFO**, 2020, São José dos Campos – SP, p. 168–173.

FERREIRA, K.R.; QUEIROZ, G.R.; VINHAS, L.; MARUJO, R.F.B.; SIMOES, R.E.O.; PICOLI, M.C.A.; CAMARA, G.; CARTAXO, R.; GOMES, V.C.F.; SANTOS, L.A.; SANCHEZ, A.H.; ARCANJO, J.S.; FRONZA, J.G.; NORONHA, C.A.; COSTA, R.W.; ZAGLIA, M.C.; ZIOTI, F.; KORTING, T.S.; SOARES, A.R. CHAVES, M.E.D.; FONSECA, L.M.G. Earth Observation Data Cubes for Brazil: Requirements, Methodology and Products. **Remote Sensing**, v. 12, n. 24, 2020.

GIULIANI, G.; CHATENOUX, B.; DE BONO, A.; RODILA, D.; RICHARD, J.-P.; ALLENBACH, K.; DAO, H.; PEDUZZI, P. Building an Earth Observations Data Cube: lessons learned from the Swiss Data Cube (SDC) on generating Analysis Ready Data (ARD). **Big Earth Data**, n. 1-2, v. 1, p. 100–117, 2017 DOI. 10.1080/20964471.2017.1398903.

GIULIANI, G.; MASÓ, J.; MAZZETTI, P.; NATIVI, S.; ZABALA, A. Paving the Way to Increased Interoperability of Earth Observations Data Cubes. **Data**, v. 4, n. 3, 2019.

GOMES, V.; QUEIROZ, G.; FERREIRA, K. An Overview of Platforms for Big Earth Observation Data Management and Analysis. **Remote Sensing**, n. 8, v. 12, 2020.

HOYER, S.; HAMMAN, J. xarray: N-D labeled Arrays and Datasets in Python. **Journal of Open Research Software**, v. 5, p.10, 2017.

KILLOUGH, B. Overview of the Open Data Cube Initiative. In: **IGARSS - IEEE International Geoscience and Remote Sensing Symposium**, Valence, Spain, 2018.

LEUTNER, B.; HORNING, N. RStoolbox: Tools for Remote Sensing Data Analysis. **R Package 0.2.6**, 2015.

LEWIS, A.; OLIVER, S.; LYMBURNER, L.; EVANS, B.; WYBORN, L.; MUELLER, N.; RAEVKSI, G.; HOOKE, J.; WOODCOCK, R.; SIXSMITH, J.; WU, W.; TAN, P.; LI, F.; KILLOUGH, B.; MINCHIN, S.; ROBERTS, D.; AYERS, D.; BALA, B.; DWYER, J.; ET AL. The Australian Geoscience Data Cube — Foundations and lessons learned. **Remote Sensing of Environment**, v. 202, p. 276-292, 2017. DOI. 10.1016/j.rse.2017.03.015.

NATIVI, S.; MAZZETTI, P.; CRAGLIA, M. A view-based model of data-cube to support big earth data systems interoperability. **Big Earth Data**, n. 1–2, v. 1, p. 75–99, 2017.

ODC. Open Data Cube. Available in: <https://www.opendatacube.org>. Accessed on: 25 jan. 2021.

PEDREGOSA, F.; VAROQUAUX, G.; GRAMFORT, A.; MICHEL, V.; THIRION, B.; GRISEL, O.; BLONDEL, M.; PRETTENHOFER, P.; WEISS, R.; DUBOURG, V.; VANDERPLAS, J.; PASSOS, A.; COURNAPEAU, D.; BRUCHER, M.; PERROT, M.; DUCHESNAY, E. Scikit-learn: Machine Learning in Python. **Journal of Machine Learning Research**, v. 12, p. 2825–2830, 2011.

SOUSA, D.; SMALL, C. Global cross-calibration of Landsat spectral mixture models. **Remote Sensing of Environment**, v. 192, p. 139–149, 2017.

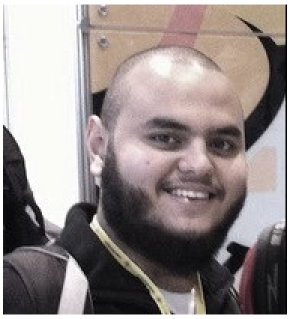
SANCHEZ, A.; PICOLI, M.; ANDRADE, P. R.; SIMÕES, R.; SANTOS, L.; CHAVES, M.; BEGOTTI, R.; CAMARA, G. Land Cover Classifications of Clear-cut Deforestation Using Deep Learning. **Proceedings XX GEOINFO**, 2019, São José dos Campos – SP, p. 48-56.

SIMOES, R.; PICOLI, M.C.A.; CAMARA, G; et al. Land use and cover maps for Mato Grosso State in Brazil from 2001 to 2017. **Scientific Data**, n. 1, v. 7, 2020. DOI. 10.1038/s41597-020-0371-4

SHIMABUKURO, Y.; PONZONI, F. **Spectral Mixture for Remote Sensing**. 2019.

SOILLE, P.; BURGER, A.; DE MARCHI, D.; KEMPENEERS, P.; RODRIGUEZ, D.; SYRRIS, V.; VASILEV, V. A versatile data-intensive computing platform for information retrieval from big geospatial data. **Future Generation Computer Systems**. v. 81, p. 30–40, 2018.

Main Author's Biography



Felipe Menino Carlos was born in September 1997 in São José dos Campos - SP. He has a degree in Systems Analysis from FATEC Jessen Vidal, where he worked with Deep Learning techniques for assistive technologies development. Currently, he is a master's student in Applied Computing at INPE, where he works on Open Reproducible Science in Earth Observation Research.