Package 'CropScapeR'

January 20, 2025

Type Package

Title Access Cropland Data Layer Data via the 'CropScape' Web Service

Version 1.1.5

Description Interface to easily access Cropland Data Layer (CDL) data for any area of interest via the 'CropScape' <https://nassgeodata.gmu.edu/CropScape/> web service.

Encoding UTF-8

License GPL (>= 2)

LazyData true

RoxygenNote 7.1.1

Depends R (>= 3.6.0)

Imports dplyr (>= 0.8.3), magrittr (>= 1.5), raster (>= 3.0), sf (>= 0.8), data.table (>= 1.12.8), httr (>= 1.4.1), RJSONIO (>= 1.3), utils (>= 3.6.1)

NeedsCompilation no

Author Bowen Chen [aut, cre] (<https://orcid.org/0000-0003-0370-2756>), Benjamin Gramig [ctb], Taro Mieno [ctb]

Maintainer Bowen Chen <bwchen0719@gmail.com>

Repository CRAN

Date/Publication 2023-01-30 15:30:11 UTC

Contents

GetCDLComp *Request data on land cover changes over time*

Description

A function that requests data on land cover changes over time from the CropScape. This function implements the GetCDLComp service provided by the CropScape https://nassgeodata.gmu.edu/CropScape.

Usage

```
GetCDLComp(
  aoi,
  year1,
  year2,
  type,
  mat = TRUE,
  crs = NULL,
  tol_time = 20,
  manual_try = TRUE
)
```

Arguments

aoi	Area of interest. Can be a 5-digit FIPS code of a county, 2-digit FIPS code of a state, four corner points or an sf object that defines a rectangle (or a box) area, multiple coordinates that defines a polygon, or a URL of an compressed ESRI shapefile.
year1	Year 1. Should be a 4-digit numeric value.
year2	Year 2. Should be a 4-digit numeric value.
type	Type of the selected AOI. 'f' for state or county, 'b' for box area, 'ps' for polygon, 's' for ESRI shapefile.
mat	TRUE/FALSE. If TRUE, return a data frame. If FALSE (default), return a raster tif file.
crs	Coordinate system. NULL if use the default coordinate system (i.e., Albers projection); Use '+init=epsg:4326' for longitude/latitude.
tol_time	Number of seconds to wait for a response until giving up. Default is 20 seconds.
manual_try	True (default) for trying calculating land cover changes using the manualrotate function. If False, no attempt is made.

Details

Land cover changes are obtained by first merging two raster files in two different years (year1 and year 2) together based on the geographic coordinates, and then count the number of pixels (or grids)

by types of land cover changes, such as corn to soybeans. The process is done in the CropScape server (the default option) or inside the GetDataComp function.

To obtain land cover chaneg data, the raster objects in two different years must have the same spatial resolutions and identical coordinates to be directly merged together. The CropScape server does internal checks on this and would report an error if the two rasters cannot be directly merged together due to unequal spatial resolution or different coordinates. However, the GetCDLComp function allows users to obtain land cover changes from two raster files that have different resolutions. This is achieved by resample the raster data using the nearest neighbor resampling technique such that both rasters have the same resolutions (the finer resolution raster is downscaled to lower resolution). Then, the resampled data are processed automatically to get data on land cover changes (this is done by using the manualrotate function). This feature is useful when dealing with the rasters in 2006 and 2007, which are at 56-meter resolution. While the rasters in other years are at 30-meter resolution. Also note that the resampling process will lead to sampling errors. Whenever the manual calculation of land cover changes is used, a warning message will show up to alert users. If without warning, the data are directly from the CropScape GetCDLComp service.

In rare cases, the CropScape server fails to generate land cover change data even without the issue of unequal spatial resolution. A common issue is mismatch in data sizes: the raster objects in two years have different pixel numbers. It is unclear that why this would happen. Nevertheless, when there is data mismatch, the GetCDLComp function will attempt to calculate for land cover change manually using the manual_rotate function. Data associated with the unmatched coordinates are discarded at the merging process. Again, a warning message will show up to alert users if manual_rotate function is used. If no coordinates can be matched, the manual_rotate function would also fail to get land cover change data. In this case, a warning message will show up to alert users.

The usage of this function is similar to the GetCDLData function. Please see the help page of the GetCDLData function for details. Note that the aoi cannot be a single point here.

Value

The function returns a data table or a raster file.

Examples

```
# Example 1. Retrieve data for the Champaign county in Illinois (FIPS = 17109) in 2017-2018.
data <- GetCDLComp(aoi = '17019', year1 = 2017, year2 = 2018, type = 'f')
head(data, 5)

# Example 2. Retrieve data for a polygon (a triangle) defined by three coordinates in 2017-2018.
aoi <- c(175207,2219600,175207,2235525,213693,2219600)
data <- GetCDLComp(aoi = aoi, year1 = 2017, year2 = 2018, type = 'ps')
head(data, 5)

# Example 3. Retrieve data for a rectangle box defined by four corner points in 2018.
data <- GetCDLComp(aoi = c(130783,2203171,153923,2217961), year1 = 2017, year2 = 2018, type = 'b')
head(data, 5)
```

GetCDLData *Request the CDL data from the CropScape*

Description

A function that requests the CDL data for any Area of Interests (AOI) in a given year from the CropScape. This function implements the GetCDLData service provided by the CropScape: https://nassgeodata.gmu.edu/CropScape.

Usage

```
GetCDLData(
  aoi = NULL,
  year = NULL,
  type = NULL,
  format = "raster",
  crs = NULL,
  tol_time = 20,
  save_path = NULL,
  readr = TRUE
)
```

Arguments

aoi	Area of interest. Can be a 5-digit FIPS code of a county, 2-digit FIPS code of a state, four corner points or an sf object that defines a rectangle (or a box) area, multiple coordinates that defines a polygon, a single coordinate that defines a point, or a URL of an compressed ESRI shapefile. See details.
year	Year of data. Should be a 4-digit numeric value.
type	Type of the selected AOI. 'f' for state or county, 'b' for box area, 'ps' for polygon, 'p' for a single coordinate, 's' for ESRI shapefile.
format	Format of the output. 'raster' for raster object, 'table' for a data table, and 'sf' for a sf object.
crs	Coordinate system. NULL if use the default coordinate system (i.e., Albers projection); Use '+init=epsg:4326' for longitude/latitude.
tol_time	Number of seconds to wait for a response until giving up. Default is 20 seconds.
save_path	Path (including the file name with the suffix: '.tif') to save the TIF file. If a path is provided, the TIF file will be saved in the computer in the specified directory. Default: NULL.
readr	Read the raster data into R. Default is TRUE. If FALSE, only the tif file is saved at save_path.

Details

The GetCDLData function implements the data request in two steps. First, the function sends data requests to the CropScape online server using the GET function from the httr package. Second, the function reads the requested data into R using the raster function from the raster package. By default, the data returned from the CropScape are in the raster-based GeoTIFF file format. Users can choose to save the raw GeoTIFF data into their local drives.

Users should at least specify aoi, year, and type to make valid data requests. aoi, or area of interest, refers to the area to make data request. An aoi can be a state/county defined by a 2-digit/5-digit FIPS code, a box defined by four corner points, a polygon defined by multiple coordinates, a single point defined by a coordinate, or a custom area defined by an ESRI shapefile.

If the type of aoi is a box, users should specify aoi as a numeric vector with four elements that represent corner points of the box. The format of the box should be (minimum x, minimum y, maximum x, maximum y). For example, if latitude/longitude is used, users should specify the aoi as (Lower longitude, Lower latitude, Higher longitude, Higher latitude). Note that for box type aoi, users can also use a sf object as the input of aoi. In this case, the GetCDLData will automatically extract bounding box points from the sf object and then make data request. The sf object must contain the crs information. If the type of aoi is a polygon, users should specify aoi as a numeric vector with at least 6 elements (corresponding to multiple points). The format is (x1, y2, x2, y2, ..., xn, yn). The polygon can take any shape. If the type of aoi is a custom area, users must specify aoi as a URL of a compressed ESRI shapefile. The .shp, .shx, .dbf, and .prj files must all be compressed with no subdirectories in a single ZIP file. In cases that the compressed shapefile is saved in the local disk, this shapefile needs to be published to a website URL. See more detailed instructions from here: https://github.com/cbw1243/CropScapeR

The GetCDLData function provides some additional functionalities that might benefit the users. First, it can recognize data requests made in any coordinate system. The default coordinate system used by the CDL is a projected coordinate system called Albers projection (or Albers equal-area conic projection). Users can specify an alternative coordinate system, such as latitude/longitude, by changing the crs value. As an exception, this functionality is unavailable when the requested data type is 's' (a shapefile). This is because the zipped shapefile is directly sent to CropScape, and it cannot be processed before sending the request. If a shapefile is used, users must ensure that the shapefile has the Albers projection system. Second, the tol_time argument specifies the upper time limit for making the data request. This is useful particularly when the CropScape server has issues with responding to the data request (e.g., system maintenance). It is possible that the CropScape server takes a long time before sending back a message saying that the data are not available. The default time limit is 20 seconds. Third, users can choose to save the raster TIF file in their local disks when requesting the CDL data. This can be done simply by specifying a directory name in save_path. In this case, GetCDLData will first save the data and then read the saved data into R using the raster function from the raster package. For example, when letting save_path be 'C:/test.tif', the raster TIF file will be saved at the 'C' disk in the name of 'test.tif'. If save_path is NULL (default), the raster TIF file will not be saved but just read into the R environment through the raster function. Forth, users can transform the raster data into a data table by letting format = 'table' or a sf object by letting format = 'sf'. The transformation into data table is done by using the as.data.frame function from the raster package. The returned object would be a data table with the coordinates (first two columns) and numeric codes of the land cover category (third column). The coordinates are centroids of the grid cells. The transformation into 'sf' is based on functions in the 'sf' package.

The CDL website provides an EXCEL file that links the numeric codes with the land cover names.

Users can download the EXCEL file from this link https://www.nass.usda.gov/Research_and_Science/Cropland/docs/cdl_codes_names.xlsx. Alternatively, users can also use `data(linkdata)` to get the data directly from this package. Yet, be noted that `linkdata` saved in this package is not frequently updated.

Mac users might encounter the error of 'SSL certificate expired', and this error can be avoided by skipping the SSL check. Please visit the package website (https://github.com/cbw1243/CropScapeR) for details.

Value

The function returns the requested CDL data for the aoi in a given year. Format of the returned object can be a raster, a data table, or a sf, depending on users' choice.

Examples

```
# Example. Retrieve data for the Champaign county in Illinois (FIPS: 17109) in 2018.
data <- GetCDLData(aoi = 17019, year = 2018, type = 'f')
data # can plot the data using raster::plot(data)

# Same request but also save the raster data as a TIF file.
# Note: A temporary file is created to save the data using the tempfile function
data <- GetCDLData(aoi = 17019, year = 2018, type = 'f', save_path = tempfile(fileext = '.tif'))
data # can plot the data using raster::plot(data)

# Example. Retrieve data for the state of Delaware (fips: 44) in 2018.
data <- GetCDLData(aoi = 44, year = 2018, type = 'f')
data # can plot the data using raster::plot(data)

# Example. Retrieve data for a box area defined by four corner points (long/lat)
data <- GetCDLData(aoi = c(-88.2, 40.03, -88.1, 40.1), year = '2018', type = 'b',
crs = '+init=epsg:4326')
data # can plot the data using raster::plot(data)

# Example. Retrieve data for a polygon (triangle) area defined by three coordinates in 2018.
data <- GetCDLData(aoi = c(175207,2219600,175207,2235525,213693,2219600), year = 2018, type = 'ps')
data # can plot the data using raster::plot(data)

# Example. Retrieve data for a box area defined by four corner points in 2018.
data <- GetCDLData(aoi = c(130783,2203171,153923,2217961), year = '2018', type = 'b')
data # can plot the data using raster::plot(data)

# Example. Retrieve data for a single point by long/lat in 2018.
data <- GetCDLData(aoi = c(-94.6754,42.1197), year = 2018, type = 'p', crs = '+init=epsg:4326')
data

# Below uses the same point, but under the default coordinate system
data <- GetCDLData(aoi = c(108777,2125055), year = 2018, type = 'p')
data

# Visit the package webiste https://github.com/cbw1243/CropScapeR for more examples.
```

GetCDLImage *Request images of the CDL data*

Description

A function that requests images of the CDL data for an area of interests in a given year from the CropScape. This function implements the GetCDLImage service provided by the CropScape https://nassgeodata.gmu.edu/CropScape.

Usage

```
GetCDLImage(
  aoi = NULL,
  year = NULL,
  type = NULL,
  format = "png",
  crs = NULL,
  destfile = NULL,
  verbose = TRUE,
  tol_time = 20
)
```

Arguments

aoi	Area of interest. Can be a 5-digit FIPS code of a county, 2-digit FIPS code of a state, four corner points or an sf object that defines a rectangle (or a box) area, multiple coordinates that defines a polygon, or a URL of an compressed ESRI shapefile.
year	Year of data. Should be a 4-digit numeric value.
type	Type of the selected AOI. 'f' for state or county, 'b' for box area, 'ps' for polygon, 's' for ESRI shapefile.
format	Format of the image file. Can be png or kml.
crs	Coordinate system. NULL if use the default coordinate system (i.e., Albers projection); Use '+init=epsg:4326' for longitude/latitude.
destfile	A character string that specifies the directory to save the downloaded image file (e.g., 'C:/image.png'). Note that the name of the image file should be specified as well. If not providing destfile, the function will create a temporary folder to save the image file.
verbose	TRUE/FALSE. Display the directory saving the file or not.
tol_time	Number of seconds to wait for a response until giving up. Default is 20 seconds.

Details

The usage of this function is similar to the GetCDLData function. Please see the help page of the GetCDLData function for details. Note that the aoi cannot be a single point here.

Value

The function downloads an image file in png or kml format to users' computer. This function is different to `GetCDLData` that returns a raster TIF file.

`GetCDLStat` *Request summary statistics of the CDL data*

Description

A function that requests summary statistics of the CDL data for any Area of Interests (AOI) in a given year from the CropScape. This function implements the GetCDLStat services provided by the CropScape https://nassgeodata.gmu.edu/CropScape.

Usage

```
GetCDLStat(aoi = NULL, year = NULL, type = NULL, crs = NULL, tol_time = 20)
```

Arguments

`aoi`	Area of interest. Can be a 5-digit FIPS code of a county, 2-digit FIPS code of a state, four corner points or an sf object that defines a rectangle (or a box) area, multiple coordinates that defines a polygon, or a URL of an compressed ESRI shapefile.
`year`	Year of data. Should be a 4-digit numeric value.
`type`	Type of the selected AOI. 'f' for state or county, 'b' for box area, 'ps' for polygon, 's' for ESRI shapefile.
`crs`	Coordinate system. `NULL` if use the default coordinate system (i.e., Albers projection); Use '+init=epsg:4326' for longitude/latitude.
`tol_time`	Number of seconds to wait for a response until giving up. Default is 20 seconds.

Details

The usage of this function is similar to the `GetCDLData` function. Please see the help page of the `GetCDLData` function for details. Note that the `aoi` cannot be a single point here.

Value

The function returns a data frame that reports summary statistics of the CDL data for an AOI in a given year.

Examples

```
# Example 1. Retrieve data for the Champaign county in Illinois (FIPS = 17109) in 2018.
data <- GetCDLStat(aoi = 17019, year = 2018, type = 'f')
head(data, n = 5) # Show top 5 rows of retrieved data

# Example 2. Retrieve data for a polygon (a triangle) defined by three points in 2018.
data <- GetCDLStat(aoi = c(175207,2219600,175207,2235525,213693,2219600), year = 2018, type = 'ps')
head(data, n = 5)

# Example 3. Retrieve data for a rectangle box defined by three corner points in 2018.
data <- GetCDLStat(aoi = c(130783,2203171,153923,2217961), year = '2018', type = 'b')
head(data, n = 5)
```

---

linkdata *A dataset documenting the correspondence between crop names and crop names for CDL*

Description

A dataset documenting the correspondence between crop names and crop names for CDL

Usage

```
data(linkdata)
```

Format

An object of class `data.table` (inherits from `data.frame`) with 256 rows and 2 columns.

Source

https://www.nass.usda.gov/Research_and_Science/Cropland/docs/cdl_codes_names.xlsx

---

manualrotate *Manual calculation of land cover changes*

Description

The `manualrotate` function analyzes land cover changes based on two raster files from the CropScape. The analysis is done in three steps. At step 1, the two raster files are converted to data tables. At step 2, the two data tables are merged together based on their coordinates. The coordinates without macthes are discarded during the merging process. At step 3, the merged data are aggregated by counting the number of pixels for each land cover change group.

Usage

```
manualrotate(datat1, datat2)
```

Arguments

datat1	A raster file.
datat2	A raster file.

Value

The function returns a data frame.

Index