

Package: rgeeExtra (via r-universe)

May 2, 2026

Title Extensions for 'rgee'

Version 0.1.0

Description High-level API to process Google Earth Engine (GEE) raster (ee.Image and ee.ImageCollection) and vector data (ee.Geometry, ee.Feature, and ee.FeatureCollection). Popular Third-party GEE algorithms are re-coded from Javascript and Python to R.

License Apache License (>= 2)

Encoding UTF-8

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 7.2.3

Config/reticulate list(packages = list(list(package =
 ``earthengine-api")))

Imports reticulate (>= 1.31), rgee, magrittr, crayon, cli

Suggests testthat (>= 3.0.0), googledrive, googleCloudStorageR,
 geojsonio, magick, stars, sf, knitr, rmarkdown

VignetteBuilder knitr

Language en-US

BugReports <https://github.com/r-earthengine/rgeeExtra/issues>

URL <https://github.com/r-earthengine/rgeeExtra>

Config/testthat/edition 3

Config/pak/sysreqs libabsl-dev cmake libgdal-dev gdal-bin libgeos-dev make libpng-dev libuv1-
 dev libssl-dev libproj-dev python3 libsqlite3-dev libudunits2-dev

Repository <https://ropensci-champions.r-universe.dev>

Date/Publication 2023-11-23 14:17:14 UTC

RemoteUrl <https://github.com/r-earthengine/rgeeExtra>

RemoteRef HEAD

RemoteSha 7ea67d064e56e9429971d64440d8424df68f6249

Contents

rgeeExtra-package	2
ee-citation	3
ee-getdoi	4
ee-getoffset	5
ee-getscaleparams	6
ee-getstac	7
ee-pansharpen	8
ee-preprocess	9
ee-scaleandoffset	10
ee-subset	11
ee-tasseledcap	13
ee_extract_featurecollection	14
ee_get	15
ee_Image_maskClouds	16
ee_Image_matchHistogram	17
ee_Image_spectralIndex	18
ee_ImageCollection_closest	20
ee_names_feature	21
ee_names_imagecollection	21
ee_utils_gif_annotate	22
ee_utils_gif_creator	24
ee_utils_gif_save	26
extra_Initialize	28
extremeValues	29
length.ee.image.Image	30
Math-methods	31
names.ee.image.Image	32
names(x) <- value	33
Ops-methods	33
Summary-methods	35
Index	36

 rgeeExtra-package

rgeeExtra: Extra functionality for rgee

Description

Google Earth Engine (Gorelick et al., 2017) is a cloud computing platform designed for planetary-scale environmental data analysis that only can be accessed via the Earth Engine code editor, third-party web apps, and the JavaScript and Python client libraries. *rgee* is a non-official client library for R that uses *reticulate* to wrap the Earth Engine Python API and provide R users with a familiar interface, rapid development features, and flexibility to analyze data using open-source, R third-party packages.

Details

The package implements and supports:

- Math operators

Author(s)

Maintainer: Cesar Aybar <csaybar@gmail.com> ([ORCID](#))

Other contributors:

- David Montero <dm1.mont@gmail.com> ([ORCID](#)) [contributor]
- Lesly Bautista <leslyarcelly.213@gmail.com> ([ORCID](#)) [contributor]
- Marc Choisy <mchoisy@oucru.org> ([ORCID](#)) [contributor]

See Also

Useful links:

- <https://github.com/r-earthengine/rgeeExtra>
- Report bugs at <https://github.com/r-earthengine/rgeeExtra/issues>

ee-citation

Citing EE objects in publications

Description

If it exists, retrieve the citation of an EE object.

Usage

```
`ee$Image$Extra_getCitation(x)`
```

```
`ee$ImageCollection$Extra_getCitation(x)`
```

Arguments

x An EE object to get the citation from.

Value

A character with citation information.

Examples

```
## Not run:
library(rgee)
library(rgeeExtra)

ee_Initialize()
extra_Initialize()

# Retrieve citation for the first image in NASA's IMERG V06 collection
citation <- ee$ImageCollection("NASA/GPM_L3/IMERG_V06")[[1]] %>%
  ee$Image$Extra_getCitation()

# Display the citation
citation

# Fetching NASA/GPM_L3/IMERG_V06 image collection and retrieving its citation.
citation_ <- ee$ImageCollection("NASA/GPM_L3/IMERG_V06") %>%
  ee$ImageCollection$Extra_getCitation()

# Display the citation
citation_

## End(Not run)
```

ee-getdoi

Get the Digital Object Identifier (DOI) of an EE object

Description

If it exists, retrieve the DOI of an EE object.

Usage

```
`ee$Image$Extra_getDOI(x)`
`ee$ImageCollection$Extra_getDOI(x)`
```

Arguments

x An EE object to get the DOI from.

Value

A character with DOI information.

Examples

```
## Not run:
library(rgee)
library(rgeeExtra)

ee_initialize()
extra_initialize()

# Fetch DOI for first image in NASA IMERG V06 collection
doi <- ee$ImageCollection("NASA/GPM_L3/IMERG_V06")[[1]] %>%
  ee$Image$Extra_getDOI()

doi

# Retrieve and print the DOI for the NASA/GPM_L3/IMERG_V06 image collection.
doi_ <- ee$ImageCollection("NASA/GPM_L3/IMERG_V06") %>%
  ee$ImageCollection$Extra_getDOI()

doi_

## End(Not run)
```

ee-getoffset	<i>Retrieve offset parameter from EE Image and EE ImageCollections objects</i>
--------------	--

Description

Earth Engine apply a simply lossless compression technique: $IMG_Float_Values = scale * IMG_Integer_Values + offset$. `ee$Image$getOffsetParams` or `ee$ImageCollection$getOffsetParams` retrieve the offset parameter for each band of an `ee$Image`.

Usage

```
`ee$Image$Extra_getOffsetParams(x)`
`ee$ImageCollection$Extra_getOffsetParams(x)`
```

Arguments

x An `ee$Image` or an `ee$ImageCollection` object.

Value

A list with the offset parameters for each band.

Examples

```
## Not run:
library(rgee)
library(rgeeExtra)

ee_Initialize()
extra_Initialize()

# Retrieve offset parameters from the first image in NASA IMERG V06 collection
offset_params <- ee$ImageCollection("NASA/GPM_L3/IMERG_V06")[[1]] %>%
  ee$Image$Extra_getOffsetParams()

# Display offset parameters for each band.
offset_params

# Get offset parameters from NASA IMERG V06 image collection.
offset_params_ <- ee$ImageCollection("NASA/GPM_L3/IMERG_V06") %>%
  ee$ImageCollection$Extra_getOffsetParams()

# Display offset parameters for each band.
offset_params_

## End(Not run)
```

ee-getscaleparams	<i>Retrieve scale parameter from EE Image and EE ImageCollection objects.</i>
-------------------	---

Description

Earth Engine apply a simply lossless compression technique: $IMG_Float_Values = scale * IMG_Integer_Values + offset$. `ee$Image$getScaleParams` and `ee$ImageCollection$getScaleParams` retrieve the scale parameter for each band of an `ee$Image`.

Usage

```
`ee$Image$Extra_getScaleParams(x)`
`ee$ImageCollection$Extra_getScaleParams(x)`
```

Arguments

x An `ee$Image` or an `ee$ImageCollection` object.

Value

A list with the scale parameters for each band.

Examples

```

## Not run:
library(rgee)
library(rgeeExtra)

ee_Initialize()
extra_Initialize()

# Retrieve scale parameters from the first image in NASA IMERG V06 collection
scale_params <- ee$ImageCollection("NASA/GPM_L3/IMERG_V06")[[1]] %>%
  ee$Image$Extra_getScaleParams()

# Display scale parameters for each band in the image.
scale_params

# Retrieve scale parameters for the NASA IMERG V06 collection.
scale_params_ <- ee$ImageCollection("NASA/GPM_L3/IMERG_V06") %>%
  ee$Image$Extra_getScaleParams()

# Display scale parameters for each band in the image.
scale_params_

## End(Not run)

```

ee-getstac

Retrieve EE Image or EE ImageCollection STAC metadata

Description

Get the STAC of an ee\$Image or ee\$ImageCollection object.

Usage

```

`ee$Image$Extra_getSTAC(x)`

`ee$ImageCollection$Extra_getSTAC(x)`

```

Arguments

x An ee\$Image or an ee\$ImageCollection object.

Value

Return STAC metadata for each band.

Examples

```
## Not run:
library(rgee)
library(rgeeExtra)

ee_initialize()
extra_initialize()

# Retrieve STAC metadata for the first image in NASA's GPM L3 IMERG V06 collection
stac_metadata <- ee$ImageCollection("NASA/GPM_L3/IMERG_V06")[[1]] %>%
  ee$Image$Extra_getSTAC()

stac_metadata

# Retrieve STAC metadata from NASA's IMERG V06 image collection.
stac_metadata_ <- ee$ImageCollection("NASA/GPM_L3/IMERG_V06") %>%
  ee$ImageCollection$Extra_getSTAC()

stac_metadata_

## End(Not run)
```

ee-pansharpen *Apply panchromatic sharpening to an ee\$Image or an ee\$ImageCollection*

Description

Apply panchromatic sharpening to an ee\$Image. Optionally, run quality assessments between the original and sharpened Image to measure spectral distortion and set results as properties of the sharpened Image.

Usage

```
`ee$Image$Extra_panSharpen(x, ...)`
```

Arguments

x	An ee\$Image object, the image to sharpen.
...	Additional arguments including sharpening method, quality assessments, and parameters for ee.Image.reduceRegion(). See details for more information.

Details

The ... argument can include the following:

- methodCharacter. The sharpening algorithm to apply. Options include “SFIM”, “HPFA”, “PCS”, and “SM”. Default is “SFIM”.

- `qaCharacter`. One or more quality assessment names to apply after sharpening, such as “MSE”, “RASE”, “UIQP”, etc.
- `geometry`, `maxPixels`, `bestEffort`, etc. Arguments passed to `ee.Image.reduceRegion()` during PCS sharpening and quality assessments.

For the PCS method, additional parameters for `ee.Image.reduceRegion()` can be specified, such as `geometry`, `maxPixels`, `bestEffort`, etc.

Value

The Image with all sharpenable bands sharpened to the panchromatic resolution and quality assessments run and set as properties.

Examples

```
## Not run:
library(rgee)
library(rgeeExtra)

ee_initialize()
extra_initialize()

img <- ee$Image("LANDSAT/LC09/C02/T1_TOA/LC09_047027_20230815")
img_sharp <- ee$Image$Extra_panSharpen(img, method="HPFA", qa=c("MSE", "RMSE"), maxPixels=1e13)

Map$centerObject(img)
Map$addLayer(img_sharp, list(bands=c("B4", "B3", "B2"))) |
Map$addLayer(img, list(bands=c("B4", "B3", "B2")))

## End(Not run)
```

ee-preprocess

Automated EE Image or EE ImageCollection preprocessing

Description

Preprocessing of `ee$Image` or `ee$ImageCollection` objects. This function performs the following tasks:

- **Cloud Masking:** Remove cloud and cloud shadow pixels. See `ee$Image$cloudmask`.
- **Decompress:** Convert integer pixels to float point numbers.

Usage

```
`ee$Image$Extra_preprocess(x, ...)`

`ee$ImageCollection$Extra_preprocess(x, ...)`
```

Arguments

x An ee\$Image or an ee\$ImageCollection object.
 ... Arguments to pass to ee\$Image\$cloudmask.

Value

An ee\$Image or ee\$ImageCollection object

Examples

```
## Not run:
library(rgee)
library(rgeeExtra)

ee_initialize()
extra_initialize()

# Load a Sentinel-2 image and apply automated preprocessing.
img <- ee$Image("COPERNICUS/S2_SR/20170328T083601_20170328T084228_T35SQA") %>%
  ee$Image$Extra_preprocess()

# Load and preprocess Sentinel-2 SR image collection.
ic <- ee$ImageCollection$Dataset$COPERNICUS_S2_SR %>%
  ee$ImageCollection$Extra_preprocess()

## End(Not run)
```

ee-scaleandoffset *Automatic decompression of EE Image or EE ImageCollection objects*

Description

Earth Engine apply a simply lossless compression technique: $IMG_Float_Values = scale * IMG_Integer_Values + offset$. ee\$Image\$scaleAndOffset or ee\$ImageCollection\$scaleAndOffset backs the integer pixel values to float point number.

Usage

```
`ee$Image$Extra_scaleAndOffset(x)`
`ee$ImageCollection$Extra_scaleAndOffset(x)`
```

Arguments

x An ee\$Image or an ee\$ImageCollection object.

Value

An ee\$Image or an ee\$ImageCollection with float pixel values.

Examples

```
## Not run:
library(rgee)
library(rgeeExtra)

ee_initialize()
extra_initialize()

# Adjust first image in NASA IMERG V06 for scale and offset.
adjusted_image <- ee$ImageCollection("NASA/GPM_L3/IMERG_V06")[[1]] %>%
  ee$Image$Extra_scaleAndOffset()

# Adjust Sentinel-2 SR images for scale and offset.
adjusted_images <- ee$ImageCollection("COPERNICUS/S2_SR") %>%
  ee$ImageCollection$Extra_scaleAndOffset()

## End(Not run)
```

ee-subset

Extract or replace parts of ee\$Image and ee\$ImageCollection

Description

Extract or replace parts of ee\$Image and ee\$ImageCollection
 Extract or replace parts of and ee\$ImageCollection

Usage

```
## S3 method for class 'ee.image.Image'
x[[index]]

## S3 replacement method for class 'ee.image.Image'
x[index] <- value

## S3 replacement method for class 'ee.image.Image'
x[[index]] <- value

## S3 method for class 'ee.imagecollection.ImageCollection'
x[[index]]

## S3 replacement method for class 'ee.imagecollection.ImageCollection'
x[[index]] <- value
```

Arguments

x	ee\$ImageCollection or ee\$Image.
index	Integer. Index specifying elements to extract or replace.
value	ee\$ImageCollection or ee\$Image to replace in.

Value

Bands of ee\$Image or an ee\$Image

An EE.ImageCollection

Examples

```
## Not run:
library(rgee)
library(rgeeExtra)
library(sf)

ee_Initialize(gcs = TRUE, drive = TRUE)

# Define a Image or ImageCollection: Terraclimate
terraclimate <- ee$ImageCollection("IDAHO_EPSCOR/TERRACLIMATE") %>%
  ee$ImageCollection$filterDate("2001-01-01", "2002-01-01")

# Define temperature Vis parameters
maximumTemperatureVis <- list(
  min = -300.0,
  max = 300.0,
  palette = c(
    '1a3678', '2955bc', '5699ff', '8dbae9', 'acd1ff', 'caebff', 'e5f9ff',
    'fdffb4', 'ffe6a2', 'ffc969', 'ffa12d', 'ff7c1f', 'ca531a', 'ff0000',
    'ab0000'
  )
)

Map$setCenter(71.72, 52.48, 2)
tnames <- names(terraclimate[[2]])
m1 <- Map$addLayer(terraclimate[[2]][["tmmx"]], maximumTemperatureVis)

terraclimate[[2]] <- terraclimate[[2]]*1.4
names(terraclimate[[2]]) <- tnames
m2 <- Map$addLayer(terraclimate[[2]][["tmmx"]], maximumTemperatureVis)
m1 | m2

## End(Not run)
## Not run:
library(rgee)
library(rgeeExtra)
library(sf)

ee_Initialize(gcs = TRUE, drive = TRUE)
extra_Initialize()
# Define a Image or ImageCollection: Terraclimate

## End(Not run)
```

ee-tasseledcap	<i>Calculates tasseled cap brightness, wetness, and greenness components for an EE Image or EE ImageCollection</i>
----------------	--

Description

Tasseled cap transformations are applied using coefficients published for these supported platforms:

- 1. Sentinel-2 MSI Level 1C (1)
- 2. Landsat 9 OLI-2 SR (2)
- 3. Landsat 9 OLI-2 TOA (2)
- 4. Landsat 8 OLI SR (2)
- 5. Landsat 8 OLI TOA (2)
- 6. Landsat 7 ETM+ TOA (3)
- 7. Landsat 5 TM Raw DN (4)
- 8. Landsat 4 TM Raw DN (5)
- 9. Landsat 4 TM Surface Reflectance (6)
- 10. MODIS NBAR (7)

Usage

```
`ee$Image$Extra_tasseledCap(x)`
```

Arguments

x	ee\$Image to calculate tasseled cap components for. Must belong to a supported platform.
---	--

Details

- 1. Shi, T., & Xu, H. (2019). Derivation of Tasseled Cap Transformation Coefficients for Sentinel-2 MSI At-Sensor Reflectance Data. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 1–11. doi:10.1109/jstars.2019.2938388
- 2. Zhai, Y., Roy, D.P., Martins, V.S., Zhang, H.K., Yan, L., Li, Z. 2022. Conterminous United States Landsat-8 top of atmosphere and surface reflectance tasseled cap transformation coefficients. *Remote Sensing of Environment*, 274(2022). doi:10.1016/j.rse.2022.112992
- 3. Huang, C., Wylie, B., Yang, L., Homer, C. and Zylstra, G., 2002. Derivation of a tasselled cap transformation based on Landsat 7 at-satellite reflectance. *International journal of remote sensing*, 23(8), pp.1741-1748.
- 4. Crist, E.P., Laurin, R. and Cicone, R.C., 1986, September. Vegetation and soils information contained in transformed Thematic Mapper data. In *Proceedings of IGARSS'86 symposium* (pp. 1465-1470). Paris: European Space Agency Publications Division.

- 5. Crist, E.P. and Cicone, R.C., 1984. A physically-based transformation of Thematic Mapper data—The TM Tasseled Cap. IEEE Transactions on Geoscience and Remote sensing, (3), pp.256-263.
- 6. Crist, E.P., 1985. A TM tasseled cap equivalent transformation for reflectance factor data. Remote sensing of Environment, 17(3), pp.301-306.
- 7. Lobser, S.E. and Cohen, W.B., 2007. MODIS tasselled cap: land cover characteristics expressed through transformed MODIS data. International Journal of Remote Sensing, 28(22), pp.5079-5101.

Value

ee\$Image with the tasseled cap components as new bands.

Examples

```
## Not run:
library(rgee)
library(rgeeExtra)

ee_initialize()
extra_initialize()

img <- ee$Image("LANDSAT/LT05/C01/T1/LT05_044034_20081011")
img <- ee$Image$Extra_tasseledCap(img)
names(img)

## End(Not run)
```

ee_extract_featurecollection

Extract parts of and EE FeatureCollection

Description

Extract parts of and EE FeatureCollection

Usage

```
## S3 method for class 'ee.featurecollection.FeatureCollection'
x[[index]]
```

Arguments

x ee\$FeatureCollection.
 index Integer. Index specifying elements to extract or replace.

Value

An ee\$FeatureCollection

Examples

```
## Not run:
library(rgee)
library(rgeeExtra)
library(sf)

ee_Initialize(gcs = TRUE, drive = TRUE)
extra_Initialize()

# Define a Image or ImageCollection: Terraclimate
fc_tiger <- ee$FeatureCollection('TIGER/2016/Roads')
fc_tiger_subset <- fc_tiger[[1:100]]
Map$centerObject(fc_tiger_subset)
Map$addLayer(fc_tiger_subset)

## End(Not run)
```

ee_get	<i>Return the element at a specified position in an Earth Engine Image or ImageCollection</i>
--------	---

Description

Return the element at a specified position in an Earth Engine Image or ImageCollection

Usage

```
ee_get(ee_c, index = 0)
```

Arguments

ee_c	ImageCollection or FeatureCollection.
index	Numeric. Specified position.

Value

Depending of ee_c can return either an ee\$FeatureCollection or ee\$ImageCollection.

Examples

```
## Not run:
library(rgee)
library(sf)

ee_Initialize()

nc <- st_read(system.file("shape/nc.shp", package = "sf")) %>%
  st_transform(4326) %>%
```

```

sf_as_ee()

ee_s2 <- ee$ImageCollection("COPERNICUS/S2")$
  filterDate("2016-01-01", "2016-01-31")$
  filterBounds(nc)

ee_s2$size()$getInfo() # 126

# Get the first 5 elements
ee_get(ee_s2, index = 0:4)$size()$getInfo() # 5

## End(Not run)

```

ee_Image_maskClouds *Masks clouds and shadows*

Description

Masks clouds and shadows in an ee.Image. Valid just for Surface Reflectance products. This function may mask water as well as clouds for the Sentinel-3 Radiance product.

Usage

```
`ee$Image$Extra_maskClouds(x, ...)`
```

Arguments

x	An ee\$Image to be processed for cloud and shadow masking.
...	Additional arguments for cloud and shadow masking. See details for more information.

Details

The ... argument can include the following:

- methodCharacter. The method to mask clouds. Options: "cloud_prob" and "qa".
- probNumeric. Cloud probability threshold, between 0 and 100. Valid for 'cloud_prob' method. Default 60.
- maskCirrusLogical. Whether to mask cirrus clouds. Valid for 'qa' method. Default TRUE.
- maskShadowsLogical. Whether to mask cloud shadows. Default TRUE.
- scaledImageLogical. If TRUE, scale pixel values to <0,1>. Default FALSE.
- darkNumeric. NIR threshold for potential cloud shadows, between 0-1. Default 0.15.
- cloudDistNumeric. Max distance in meters to search for cloud shadows from cloud edges. Default 1000m.
- bufferNumeric. Distance in meters to dilate cloud and shadow objects. Default 250m.
- cdiNumeric. Cloud Displacement Index threshold, between <-1, 1>. Default NULL.

For more information on parameters and methods, refer to relevant cloud masking literature and tutorials.

Value

ee\$Image with a Cloud-shadow masked image.

Examples

```
## Not run:
library(rgee)
library(rgeeExtra)

ee_Initialize()
extra_Initialize()

img <- ee$ImageCollection("COPERNICUS/S2_SR") %>%
  ee$ImageCollection$first() %>%
  ee$Image$Extra_maskClouds(prob = 75,buffer = 300,cdi = -0.5)

names(img)

## End(Not run)
```

ee_Image_matchHistogram

Adjust the image's histogram to match a target image

Description

Matches the histogram of one image (source) to that of another image (target).

Usage

```
`ee$Image$Extra_matchHistogram(x, ...)`
```

Arguments

x	ee\$Image to adjust.
...	Additional arguments for histogram matching. See details for more information.

Details

The ... argument can include the following:

- targetee\$Image. The target image to match.
- bandsDictionary. Band names to match, with source bands as keys and target bands as values.
- geometryee\$Geometry. The region to match histograms in that overlaps both images. Default is NULL.
- maxBucketsInteger. The maximum number of buckets to use when building histograms. Default 256.

These parameters allow for detailed customization of the histogram matching process.

Value

The adjusted image containing the matched source bands.

Examples

```
## Not run:
library(rgee)
library(rgeeExtra)

ee_Initialize()
extra_Initialize()

source <- ee$Image("LANDSAT/LC08/C01/T1_TOA/LC08_047027_20160819")
target <- ee$Image("LANDSAT/LE07/C01/T1_TOA/LE07_046027_20150701")
bands <- list("B4"="B3", "B3"="B2", "B2"="B1")

matched <- ee$Image$Extra_matchHistogram(source, target, bands)

names(matched)

## End(Not run)
```

ee_Image_spectralIndex

Spectral Indices Computation

Description

Computes one or more spectral indices for an ee\$Image or an ee\$ImageCollection object.

Usage

```
`ee$Image$Extra_spectralIndex(x, ...)`
```

Arguments

x	An ee\$Image or an ee\$ImageCollection to compute indices on.
...	Additional arguments passed to the underlying spectral index computation function. See details for a complete list of possible arguments.

Details

The ... argument can include any of the following:

- indexCharacter. Index or list of indices to compute. Options include 'vegetation', 'burn', 'water', 'snow', 'drought', 'urban', 'kernel', and 'all'. Default 'NDVI'.
- GNumeric. Gain factor for 'EVI'. Default 2.5.
- C1, C2Numerics. Coefficients for aerosol resistance in 'EVI'. Defaults are 6.0 and 7.5.

- LNumeric. Canopy background adjustment for 'EVI' or 'SAVI'. Default 1.0.
- cexpNumeric. Coefficient for 'OCVI'. Default 1.16.
- nexpNumeric. Exponent for 'GDVI'. Default 2.0.
- alphaNumeric. Weighting coefficient for 'WDRVI'. Default 0.1.
- slope, interceptNumerics. Soil line slope and intercept. Defaults are 1.0 and 0.0.
- gammaNumeric. Weighting coefficient for 'ARVI'. Default 1.0.
- kernelCharacter. Kernel type for kernel indices. Options are 'linear', 'RBF', and 'poly'. Default 'RBF'.
- sigmaCharacter or Numeric. Length-scale parameter for RBF kernel. Default '0.5 * (a + b)'.
- pNumeric. Kernel degree for polynomial kernel. Default 2.0.
- cNumeric. Free parameter for polynomial kernel. Default 1.0.
- onlineLogical. Whether to retrieve the most recent list of indices online. Default FALSE.
- dropLogical. If TRUE, drop the image bands after calculation. Default TRUE.

For a complete list of indices and their parameters, please refer to the [spectral indices documentation](#).

Value

An ee\$Image or an ee\$ImageCollection with the computed spectral index, or indices, as new bands.

Examples

```
## Not run:
library(rgee)
library(rgeeExtra)

ee_Initialize()
extra_Initialize()

s2_indices <- ee$ImageCollection("COPERNICUS/S2_SR") %>%
  ee$ImageCollection$first() %>%
  ee$Image$Extra_preprocess() %>%
  ee$Image$Extra_spectralIndex(c("NDVI", "SAVI"))

names(s2_indices)
# "NDVI" "SAVI"

## End(Not run)
```

 ee_ImageCollection_closest

Get the temporal nearest image

Description

Gets the closest ee\$Image to a specified date from an ee\$ImageCollection.

Usage

```
`ee$ImageCollection$Extra_closest(x, ...)`
```

Arguments

x	ee\$ImageCollection from which to get the closest image to the specified date.
...	Additional arguments for finding the nearest image. See details for more information.

Details

The ... argument can include the following:

- dateee\$Date or R date object. Date of interest for searching the closest image.
- toleranceNumeric. Filters the collection within a range of (date - tolerance, date + tolerance). Default 1.
- unitCharacter. Units for tolerance. Options include "year", "month", "week", "day", "hour", "minute", "second". Default "month".

These parameters allow for precise control over how the closest image is determined.

Value

An ee\$Image closest to the specified date.

Examples

```
## Not run:
library(rgee)
library(rgeeExtra)

ee_initialize()

roi <- ee$Geometry$Point(c(-79, -12))
ee$ImageCollection$Dataset$MODIS_006_MCD12Q1 %>%
  ee$ImageCollection$Extra_closest("2020-10-15", 2, "year") %>%
  ee$ImageCollection$first() %>%
  Map$addLayer()

## End(Not run)
```

ee_names_feature	<i>Names of Earth Engine FeatureCollection object</i>
------------------	---

Description

Get the properties names of FeatureCollection object

Usage

```
## S3 method for class 'ee.feature.Feature'  
names(x)
```

Arguments

x an EE FeatureCollection object.

Value

A vector representing the property names of the ee\$FeatureCollection object

Examples

```
## Not run:  
library(rgee)  
library(rgeeExtra)  
  
ee_initialize()  
extra_initialize()  
  
fc <- ee$FeatureCollection('WRI/GPPD/power_plants')  
fc$propertyNames()$getInfo()  
  
## End(Not run)
```

ee_names_imagecollection	<i>Names of Earth Engine ImagesCollection properties</i>
--------------------------	--

Description

Get the names of the properties of an Earth Engine ImageCollection object.

Usage

```
## S3 method for class 'ee.imagecollection.ImageCollection'  
names(x)
```

Arguments

x an EE ImageCollection object.

Value

A vector with the property names of ee\$ImageCollection object

Examples

```
## Not run:
library(rgeeExtra)
library(rgee)

ee_initialize()     # Initialize the Google Earth Engine API connection
extra_initialize() # Initialize the extended functionalities of rgeeExtra

ic <- ee$ImageCollection("COPERNICUS/S2_SR") %>%
  ee$ImageCollection$filterDate("1999-01-01", "1999-01-02")
names(ic)

## End(Not run)
```

ee_utils_gif_annotate *Add text to a GIF*

Description

Add text to a GIF (magick-image object). This function is a wrapper around [image_annotate](#).

Usage

```
ee_utils_gif_annotate(
  image,
  text,
  gravity = "northwest",
  location = "+0+0",
  degrees = 0,
  size = 20,
  font = "sans",
  style = "normal",
  weight = 400,
  kerning = 0,
  decoration = NULL,
  color = NULL,
  strokecolor = NULL,
  boxcolor = NULL
)
```

Arguments

image	magick image object returned by <code>magick::image_read()</code> or <code>magick::image_graph()</code>
text	character vector of length equal to 'image' or length 1
gravity	string with gravity value from <code>gravity_types</code> .
location	geometry string with location relative to gravity
degrees	rotates text around center point
size	font-size in pixels
font	string with font family such as "sans", "mono", "serif", "Times", "Helvetica", "Trebuchet", "Georgia", "Palatino" or "Comic Sans".
style	value of <code>style_types</code> for example "italic"
weight	thickness of the font, 400 is normal and 700 is bold.
kerning	increases or decreases whitespace between letters
decoration	value of <code>decoration_types</code> for example "underline"
color	a valid color string such as "navyblue" or "#000080". Use "none" for transparency.
strokecolor	a color string adds a stroke (border around the text)
boxcolor	a color string for background color that annotation text is rendered on.

Value

A magick-image object

Author(s)

Jeroen Ooms

See Also

Other GIF functions: `ee_utils_gif_creator()`, `ee_utils_gif_save()`

Examples

```
## Not run:
library(rgee)
library(rgeeExtra)

ee_initialize()

col <- ee$ImageCollection("JRC/GSW1_1/YearlyHistory")$map(function(img) {
  year <- img$date()$get("year")
  yearImg <- img$gte(2)$multiply(year)
  despeckle <- yearImg$connectedPixelCount(15, TRUE)$eq(15)
  yearImg$updateMask(despeckle)$selfMask()$set("year", year)
})

appendReverse <- function(col) col$merge(col$sort('year', FALSE))
```

```

# -----
# 1 Basic Animation - Ucayali Peru
# -----

bgColor = "FFFFFF" # Assign white to background pixels.
riverColor = "0D0887" # Assign blue to river pixels.

## 1.1 Create the dataset
annualCol = col$map(function(img) {
  img$unmask(0)$
  visualize(min = 0, max = 1, palette = c(bgColor, riverColor))$
  set("year", img$get("year"))
})
basicAnimation <- appendReverse(annualCol)

## 1.2 Set video arguments
aoi <- ee$Geometry$Rectangle(-74.327, -10.087, -73.931, -9.327)
videoArgs = list(
  dimensions = 600, # Max dimension (pixels), min dimension is proportionally scaled.
  region = aoi,
  framesPerSecond = 10
)

## 1.2 Download, display and save the GIF!
animation <- ee_utils_gif_creator(basicAnimation, videoArgs, mode = "wb")
get_years <- basicAnimation$aggregate_array("year")$getInfo()
animation %>%
  ee_utils_gif_annotate("Ucayali, Peru") %>%
  ee_utils_gif_annotate(get_years, size = 15, location = "+90+40",
    boxcolor = "FFFFFF") %>%
  ee_utils_gif_annotate("created using {magick} + {rgee}",
    size = 15, font = "sans", location = "+70+20") ->
  animation_wtxt
gc(reset = TRUE)
ee_utils_gif_save(animation_wtxt, path = paste0(tempfile(), ".gif"))

## End(Not run)

```

ee_utils_gif_creator *Create a GIF from an Earth Engine ImageCollection*

Description

Create an GIF (as a magick-image object) from a EE ImageCollection. Note: Animations can only be created when ImageCollections is composed by RGB or RGBA image. This can be done by mapping a visualization function onto an ImageCollection (e.g. `ic$map(function(img) img$visualize(...))`) or specifying three bands in parameters argument (See examples). [ee_utils_gif_creator](#) is a wrapper around `ee$ImageCollection$getVideoThumbURL`.

Usage

```
ee_utils_gif_creator(ic, parameters, quiet = FALSE, ...)
```

Arguments

<code>ic</code>	An ee\$ImageCollection.
<code>parameters</code>	List of parameters for visualization and animation. See details.
<code>quiet</code>	Logical. Suppress info message.
<code>...</code>	parameter(s) passed on to download.file

Details

The parameters argument is identical to visParams (See `rgee::Map$addLayer`), plus, optionally:

- **dimensions:** A number or pair of numbers in format `c(WIDTH,HEIGHT)`. Max dimensions of the thumbnail to render, in pixels. If only one number is passed, it is used as the maximum, and the other dimension is computed by proportional scaling.
- **crs:** A CRS string specifying the projection of the output.
- **crs_transform:** The affine transform to use for the output pixel grid.
- **scale:** A scale to determine the output pixel grid; ignored if both `crs` and `crs_transform` are specified.
- **region:** ee\$Geometry\$Polygon, GeoJSON or `c(E,S,W,N)`. Geospatial region of the result. By default, the whole image.
- **format:** String. The output format (only 'gif' is currently supported).
- **framesPerSecond:** String. Animation speed.

Value

A magick-image object of the specified ImageCollection.

Author(s)

Jeroen Ooms

See Also

Other GIF functions: [ee_utils_gif_annotate\(\)](#), [ee_utils_gif_save\(\)](#)

Examples

```
## Not run:
library(rgee)
library(rgeeExtra)

ee_initialize()

col <- ee$ImageCollection("JRC/GSW1_1/YearlyHistory")$map(function(img) {
```

```

year <- img$date()$get("year")
yearImg <- img$gte(2)$multiply(year)
despeckle <- yearImg$connectedPixelCount(15, TRUE)$eq(15)
yearImg$updateMask(despeckle)$selfMask()$set("year", year)
})

appendReverse <- function(col) col$merge(col$sort('year', FALSE))

# -----
# 1 Basic Animation - Ucayali Peru
# -----

bgColor = "FFFFFF" # Assign white to background pixels.
riverColor = "0D0887" # Assign blue to river pixels.

## 1.1 Create the dataset
annualCol = col$map(function(img) {
  img$unmask(0)$
  visualize(min = 0, max = 1, palette = c(bgColor, riverColor))$
  set("year", img$get("year"))
})
basicAnimation <- appendReverse(annualCol)

## 1.2 Set video arguments
aoi <- ee$Geometry$Rectangle(-74.327, -10.087, -73.931, -9.327)
videoArgs = list(
  dimensions = 600, # Max dimension (pixels), min dimension is proportionally scaled.
  region = aoi,
  framesPerSecond = 10
)

## 1.2 Download, display and save the GIF!
animation <- ee_utils_gif_creator(basicAnimation, videoArgs, mode = "wb")
get_years <- basicAnimation$aggregate_array("year")$getInfo()
animation %>%
  ee_utils_gif_annotate("Ucayali, Peru") %>%
  ee_utils_gif_annotate(get_years, size = 15, location = "+90+40",
    boxcolor = "FFFFFF") %>%
  ee_utils_gif_annotate("created using {magick} + {rgee}",
    size = 15, font = "sans", location = "+70+20") ->
  animation_wtxt
gc(reset = TRUE)
ee_utils_gif_save(animation_wtxt, path = paste0(tempfile(), ".gif"))

## End(Not run)

```

Description

Write a magick-image object as a GIF file using the magick package. This function is a wrapper around [image_write](#).

Usage

```
ee_utils_gif_save(  
  image,  
  path = NULL,  
  format = NULL,  
  quality = NULL,  
  depth = NULL,  
  density = NULL,  
  comment = NULL,  
  flatten = FALSE  
)
```

Arguments

image	magick image object returned by image_read .
path	path a file, url, or raster object or bitmap array.
format	output format such as "png", "jpeg", "gif", "rgb" or "rgba".
quality	number between 0 and 100 for jpeg quality. Defaults to 75.
depth	color depth (either 8 or 16).
density	resolution to render pdf or svg.
comment	text string added to the image metadata for supported formats.
flatten	should the image be flattened before writing? This also replaces transparency with a background color.

Value

No return value, called to write a GIF file.

Author(s)

Jeroen Ooms

See Also

Other GIF functions: [ee_utils_gif_annotate\(\)](#), [ee_utils_gif_creator\(\)](#)

Examples

```
## Not run:  
library(rgee)  
library(rgeeExtra)
```

```

ee_Initialize()

col <- ee$ImageCollection("JRC/GSW1_1/YearlyHistory")$map(function(img) {
  year <- img$date()$get("year")
  yearImg <- img$gte(2)$multiply(year)
  despeckle <- yearImg$connectedPixelCount(15, TRUE)$eq(15)
  yearImg$updateMask(despeckle)$selfMask()$set("year", year)
})

appendReverse <- function(col) col$merge(col$sort('year', FALSE))

# -----
# 1 Basic Animation - Ucayali Peru
# -----

bgColor = "FFFFFF" # Assign white to background pixels.
riverColor = "0D0887" # Assign blue to river pixels.

## 1.1 Create the dataset
annualCol = col$map(function(img) {
  img$unmask(0)$
  visualize(min = 0, max = 1, palette = c(bgColor, riverColor))$
  set("year", img$get("year"))
})
basicAnimation <- appendReverse(annualCol)

## 1.2 Set video arguments
aoi <- ee$Geometry$Rectangle(-74.327, -10.087, -73.931, -9.327)
videoArgs = list(
  dimensions = 600, # Max dimension (pixels), min dimension is proportionally scaled.
  region = aoi,
  framesPerSecond = 10
)

## 1.2 Download, display and save the GIF!
animation <- ee_utils_gif_creator(basicAnimation, videoArgs, mode = "wb")
get_years <- basicAnimation$aggregate_array("year")$getInfo()
animation %>%
  ee_utils_gif_annotate("Ucayali, Peru") %>%
  ee_utils_gif_annotate(get_years, size = 15, location = "+90+40",
    boxcolor = "FFFFFF") %>%
  ee_utils_gif_annotate("created using {magick} + {rgee}",
    size = 15, font = "sans", location = "+70+20") ->
  animation_wtxt
gc(reset = TRUE)
ee_utils_gif_save(animation_wtxt, path = paste0(tempfile(), ".gif"))

## End(Not run)

```

Description

Load extra functionality for rgee

Usage

```
extra_Initialize(quiet = FALSE)
```

Arguments

quiet Logical. Suppress info messages.

Value

TRUE if the function runs smoothly.

Examples

```
## Not run:
library(rgee)
library(rgeeExtra)

ee_Initialize() # Initialize GEE
extra_Initialize() # Extent the GEE API

## End(Not run)
```

extremeValues

Minimum and Maximum Values

Description

Returns the minimum or maximum value of an ee.Image. The return value will be an approximation if the polygon (area of the ee.Image) contains too many pixels at the native scale.

Usage

```
`ee$Image$Extra_maxValue(x, ...)`
```

```
`ee$Image$Extra_minValue(x, ...)`
```

Arguments

x ee\$Image to analyze.

... Additional arguments for specifying the mode and sampling. See details for more information.

Details

The ... argument can include the following:

- modeCharacter. Indicates the geometry over which to reduce data. Options: "Rectangle" (default) or "Points".
- sample_sizeNumeric. Number of points to be created. Relevant only if mode is "Points".

The "Rectangle" mode uses the Image system:footprint, while the "Points" mode samples points over the Image system:footprint.

Value

A number representing the minimum or maximum value.

Examples

```
## Not run:
library(rgee)
library(rgeeExtra)

ee_initialize()
extra_initialize()

image <- ee$ImageCollection$Dataset$LANDSAT_LC08_C01_T1$first()[["B1"]]
# max values
ee$Image$Extra_maxValue(image)
ee$Image$Extra_maxValue(image, mode = "Points", sample_size = 2)

# min values
ee$Image$Extra_minValue(image)
ee$Image$Extra_minValue(image, mode = "Points")

## End(Not run)
```

length.ee.image.Image *Length of an Earth Engine Image Object*

Description

Get or set the length of an Earth Engine Image.

Usage

```
## S3 method for class 'ee.image.Image'
length(x)
```

Arguments

x an EE Image Object.

Details

If a vector is shortened, extra values are discarded and when a vector is lengthened, it is padded out to its new length with `ee$Image(0)`, with band name of `zzz_rgee_NA_%02d`.

Value

A numeric value that indicate the number of bands in a `ee$Image`.

Examples

```
## Not run:
library(rgeeExtra)
library(rgee)

ee_initialize() # Initialize the Google Earth Engine API connection
extra_initialize() # Initialize the extended functionalities of rgeeExtra

ic <- ee$Image("COPERNICUS/S2_SR/20190310T105851_20190310T110327_T30TVK")
length(ic)

## End(Not run)
```

 Math-methods

Mathematical functions

Description

Mathematical functions

Usage

```
## S3 method for class 'ee.image.Image'
Math(x, ...)
```

Arguments

<code>x</code>	<code>ee\$Image</code>
<code>...</code>	Ignored

Generic mathematical functions that can be used with an `ee$Image` object as argument: `abs`, `sign`, `sqrt`, `ceiling`, `cumprod`, `cumsum`, `log`, `log10`, `log1p`, `log2`, `acos`, `floor`, `asin`, `atan`, `exp`, `expm1`, `cos`, `cosh`, `sin`, `sinh`, `tan`, and `tanh`.

Value

An `ee$Image` object

Examples

```
## Not run:
library(rgee)
library(rgeeExtra)

ee_initialize()

roi <- ee$Geometry$Point(c(-79, -12))

a <- ee$Image(1)
b <- ee$Image(2)
c <- a + b

log1p(ee$Image(10))

## End(Not run)
```

names.ee.image.Image *Names of Earth Engine Images layers (bands)*

Description

Get the names of the layers of an Earth Engine Image object.

Usage

```
`names(x)`
```

Arguments

x an EE Image object.

Value

A vector with the name of the bands.

Examples

```
## Not run:
library(rgee)
library(rgeeExtra)

ee_initialize()
extra_initialize()

img_demo <- ee$Image("COPERNICUS/S2_SR/20190310T105851_20190310T110327_T30TVK")[[1:3]]
names(img_demo)

## End(Not run)
```

names(x) <- value *Rename of Earth Engine Images layers (bands)*

Description

Set the names of the layers of an Earth Engine Image object.

Usage

```
`names(x) <- value`
```

Arguments

x an EE Image object.
value a character vector with the same length as x.

Value

An ee\$Image with its bands renamed.

Examples

```
## Not run:
library(rgee)
library(rgeeExtra)

ee_initialize()
extra_initialize()

img_demo <- ee$Image("COPERNICUS/S2_SR/20190310T105851_20190310T110327_T30TVK")[[1:3]]
names(img_demo) <- c("B01", "B02", "B03")

## End(Not run)
```

Ops-methods

Earth Engine arithmetic, logic and compare generic functions

Description

Arithmetic, logic and compare operators for computation with ee\$Image objects and numeric values.

Usage

```
## S3 method for class 'ee.image.Image'
Ops(e1, e2)
```

Arguments

e1	Numeric or ee\$Image.
e2	Numeric or ee\$Image.

Details

- **Arith:** +, -, *, /, ^, %%%, %l%, %>>% and %>>%.
- **Logic:** !, &, l.
- **Comparison:** ==, !=, >, <, <=, >=

Value

An ee\$Image object

Examples

```
## Not run:
library(rgee)
ee_initialize()

# Sum Operator
ee1 <- ee$Image(1)
ee2 <- ee$Image(2)
ee3 <- ee1 + ee2
ee_extract(ee3, ee$Geometry$Point(0, 0))

v1 <- 1
v2 <- 2
v3 <- v1 + v2
v3

# Multiple Operators
ee4 <- ee1 / 10
ee5 <- ee4 * (ee2 - 1 + ee1^2 / ee2)
ee_extract(ee5, ee$Geometry$Point(0, 0))

v4 <- v1 / 10
v5 <- v4 * (v2 - 1 + v1^2 / v2)
v5

# multi-layer object mutiplication, no recycling
ee6 <- ee1 + c(1, 5, 10)
ee_extract(ee6, ee$Geometry$Point(0, 0))

v6 <- v1 + c(1, 5, 10)
v6

## End(Not run)
```

Summary-methods

Summary Methods

Description

Summary Methods

Usage

```
## S3 method for class 'ee.image.Image'  
Summary(..., na.rm = TRUE)  
  
## S3 method for class 'ee.image.Image'  
mean(..., na.rm = TRUE)
```

Arguments

...	ee\$Image.
na.rm	Ignore.

Value

An ee\$Image object

Examples

```
## Not run:  
library(rgee)  
library(rgeeExtra)  
  
ee_initialize()  
  
roi <- ee$Geometry$Point(c(-79, -12))  
  
img_demo <- ee$Image("COPERNICUS/S2_SR/20190310T105851_20190310T110327_T30TVK")  
max(img_demo)  
  
## End(Not run)
```

Index

- * **GIF functions**
 - ee_utils_gif_annotate, 22
 - ee_utils_gif_creator, 24
 - ee_utils_gif_save, 26
- * **calibration**
 - ee-scaleandoffset, 10
- * **extremeValues**
 - extremeValues, 29
- * **package**
 - rgeeExtra-package, 2
- [<- .ee.image.Image (ee-subset), 11
- [[.ee.featurecollection.FeatureCollection (ee_extract_featurecollection), 14
- [[.ee.image.Image (ee-subset), 11
- [[.ee.imagecollection.ImageCollection (ee-subset), 11
- [[<- .ee.image.Image (ee-subset), 11
- [[<- .ee.imagecollection.ImageCollection (ee-subset), 11

- decoration_types, 23
- download.file, 25

- ee-citation, 3
- ee-getdoi, 4
- ee-getoffset, 5
- ee-getscaleparams, 6
- ee-getstac, 7
- ee-pansharpen, 8
- ee-preprocess, 9
- ee-scaleandoffset, 10
- ee-subset, 11
- ee-tasseledcap, 13
- ee_extract_featurecollection, 14
- ee_get, 15
- ee_image_getCitation (ee-citation), 3
- ee_image_getDOI (ee-getdoi), 4
- ee_Image_getOffsetParams (ee-getoffset), 5
- ee_Image_getScaleParams (ee-getscaleparams), 6
- ee_Image_getSTAC (ee-getstac), 7
- ee_Image_maskClouds, 16
- ee_Image_matchHistogram, 17
- ee_Image_panSharpen (ee-pansharpen), 8
- ee_Image_preprocess (ee-preprocess), 9
- ee_Image_scaleAndOffset (ee-scaleandoffset), 10
- ee_Image_spectralIndex, 18
- ee_Image_tasseledCap (ee-tasseledcap), 13
- ee_ImageCollection_closest, 20
- ee_ImageCollection_getCitation (ee-citation), 3
- ee_ImageCollection_getDOI (ee-getdoi), 4
- ee_ImageCollection_getOffsetParams (ee-getoffset), 5
- ee_ImageCollection_getScaleParams (ee-getscaleparams), 6
- ee_ImageCollection_getSTAC (ee-getstac), 7
- ee_ImageCollection_preprocess (ee-preprocess), 9
- ee_ImageCollection_scaleAndOffset (ee-scaleandoffset), 10
- ee_maxValue (extremeValues), 29
- ee_minValue (extremeValues), 29
- ee_names_feature, 21
- ee_names_imagecollection, 21
- ee_utils_gif_annotate, 22, 25, 27
- ee_utils_gif_creator, 23, 24, 24, 27
- ee_utils_gif_save, 23, 25, 26
- extra_Initialize, 28
- extremeValues, 29

- gravity_types, 23

- image_annotate, 22
- image_read, 27

image_write, [27](#)

length.ee.image.Image, [30](#)

magick::image_graph(), [23](#)
magick::image_read(), [23](#)
Math-methods, [31](#)
Math.ee.image.Image (Math-methods), [31](#)
mean.ee.image.Image (Summary-methods),
[35](#)

names(x) <- value, [33](#)
names.ee.feature.Feature
 (ee_names_feature), [21](#)
names.ee.image.Image, [32](#)
names.ee.imagecollection.ImageCollection
 (ee_names_imagecollection), [21](#)
names<- .ee.image.Image (names(x) <-
 value), [33](#)

Ops-methods, [33](#)
Ops.ee.image.Image (Ops-methods), [33](#)

rgeeExtra (rgeeExtra-package), [2](#)
rgeeExtra-package, [2](#)

style_types, [23](#)
Summary-methods, [35](#)
Summary.ee.image.Image
 (Summary-methods), [35](#)