Package 'glcm'

February 27, 2020

Version 1.6.5

Date 2020-02-26

Title Calculate Textures from Grey-Level Co-Occurrence Matrices (GLCMs)

Maintainer Alex Zvoleff <azvoleff@conservation.org>

Depends R (>= 2.10.0),

Imports raster, Rcpp (>= 0.11.0)

Suggests testthat (>= 0.8.0)

LinkingTo Rcpp, RcppArmadillo

Description Enables calculation of image textures (Haralick 1973) <doi:10.1109/TSMC.1973.4309314> from grey-level co-occurrence matrices (GLCMs). Supports processing images that cannot fit in memory.

License GPL (>= 3)

URL http://www.azvoleff.com/glcm

BugReports https://github.com/azvoleff/glcm/issues

LazyData true

Encoding UTF-8

RoxygenNote 7.0.2

NeedsCompilation yes

Author Alex Zvoleff [aut, cre]

Repository CRAN

Date/Publication 2020-02-26 23:30:02 UTC

R topics documented:

`glcm-package` *Calculate textures from grey-level co-occurrence matrices (GLCMs) in R*

Description

Enables calculation of image textures derived from grey-level co-occurrence matrices (GLCMs) in R. The texture calculation is coded in C++ to optimize computation time.

Author(s)

Alex Zvoleff, <azvoleff@conservation.org>

`calc_texture` *Calculates a glcm texture for use in the glcm.R script*

Description

This function is called by the `glcm` function. It is not intended to be used directly.

Usage

```
calc_texture(d, n_grey, window_dims, shift, statistics, na_opt, na_val)
```

Arguments

`d`	a matrix containing the pixels to be used in the texture calculation
`n_grey`	number of grey levels to use in texture calculation
`window_dims`	2 element list with row and column dimensions of the texture window
`shift`	a matrix where each row gives an (x, y) shift to use when computing co-occurrency matrices. Textures will be calculated for each shift, and the average over all shifts will be returned.
`statistics`	a list of strings naming the texture statistics to calculate
`na_opt`	one of "ignore", "center", or "any"
`na_val`	what value to use to fill missing values on edges or where necessary due to chosen na_opt value

Value

a list of length equal to the length of the `statistics` input parameter, containing the selected textures measures

Examples

```
# Calculate GLCM textures on a matrix using low-level calc_texture function
d <- matrix(seq(1:25), nrow=5, ncol=5, byrow=TRUE)
calc_texture(d, n_grey=25, window_dims=c(3,3),
             shift=matrix(c(1,1), nrow=1), statistics=c('variance'),
             na_opt="any", na_val=NA)
```

`expected_textures_3x3_1x1`

GLCM textures calculated in EXELIS ENVI (for testing purposes)

Description

This is the output from running a "co-occurrence measures" calculation to calculate GLCM textures in EXELIS ENVI from the `test_raster` included in the `glcm` package. The following settings were used: window size 3x3; co-occurrence shift 1 row (y in ENVI), 1 column (x in ENVI); greyscale textures to compute: mean, variance, homogeneity, contrast, dissimilarity, entropy, second moment, correlation.

See Also

`expected_textures_5x7_2x3` `expected_textures_5x3_n1xn2`

`expected_textures_5x3_n1xn2`

GLCM textures calculated in EXELIS ENVI (for testing purposes)

Description

This is the output from running a "co-occurrence measures" calculation to calculate GLCM textures in EXELIS ENVI from the `test_raster` included in the `glcm` package. The following settings were used: window size 5x3; co-occurrence shift -1 row (y in ENVI), -2 columns (x in ENVI); greyscale quantization levels 32; textures to compute: mean, variance, homogeneity, contrast, dissimilarity, entropy, second moment, correlation.

See Also

`expected_textures_3x3_1x1` `expected_textures_5x7_2x3`

expected_textures_5x7_2x3

GLCM textures calculated in EXELIS ENVI (for testing purposes)

Description

This is the output from running a "co-occurrence measures" calculation to calculate GLCM textures in EXELIS ENVI from the `test_raster` included in the `glcm` package. The following settings were used: window size 5x7; co-occurrence shift 2 rows (y in ENVI), 3 columns (x in ENVI); greyscale textures to compute: mean, variance, homogeneity, contrast, dissimilarity, entropy, second moment, correlation.

See Also

`expected_textures_3x3_1x1` `expected_textures_5x3_n1xn2`

glcm

Image texture measures from grey-level co-occurrence matrices (GLCM)

Description

This function supports calculating texture statistics derived from grey-level co-occurrence matrices (GLCMs). The default textures are calculated using a 45 degree shift. See Details for other options.

Usage

```
glcm(x, n_grey = 32, window = c(3, 3), shift = c(1, 1), statistics =
c("mean", "variance", "homogeneity", "contrast", "dissimilarity", "entropy",
"second_moment", "correlation"), min_x=NULL, max_x=NULL, na_opt="any",
na_val=NA, scale_factor=1, asinteger=FALSE)
```

Arguments

`x`	a `RasterLayer` or `matrix`
`n_grey`	number of grey levels to use in texture calculation
`window`	the window size to consider for texture calculation as a two element integer vector (number of rows, number of columns)
`shift`	a list or matrix specifying the shift to use. See Details.
`statistics`	A list of GLCM texture measures to calculate (see Details).
`min_x`	minimum value of input `RasterLayer` (optional, `glcm` will calculate if not supplied). Useful when running `glcm` over blocks of a raster.
`max_x`	maximum value of input `RasterLayer` (optional, `glcm` will calculate if not supplied). Useful when running `glcm` over blocks of a raster.

na_opt	How to handle NA values in x. Can be set to "ignore", "any" or "center". If set to "any", all textures statistics for a given pixel will be set to NA if there are any NA values in the window around that pixel. If set to "center" this will only occur if the center value is an NA. If set to "ignore", NA values in window will be ignored.
na_val	the value to use to fill NA values on edges of x where textures cannot be calculated due to the window falling outside of the image, and as necessary depending on the chosen na_opt.
scale_factor	factor by which to multiply results. Useful if rounding results to integers (see asinteger argument).
asinteger	whether to round results to nearest integer. Can be used to save space by saving results as, for example, an 'INT2S' raster.

Details

The statistics parameter should be a list, and can include any (one or more) of the following: 'mean', 'mean_ENVI', 'variance', 'variance_ENVI', 'homogeneity', 'contrast', 'dissimilarity', 'entropy', 'second_moment', and/or 'correlation'. By default all of the statistics except for "mean_ENVI" and "variance_ENVI" will be returned .

shift can be one of:

1. a two element integer vector giving the shift (Q in Gonzalez and Woods, 2008), as (number of rows, number of columns).
2. a list of integer vectors of length 2 specifying multiple (row, col) shifts over which to calculate the GLCM textures. For example: `shift=list(c(1,1),c(-1,-1))`
3. a matrix with two columns specifying, in rows, multiple (row, col) shifts over which to calculate the GLCM textures. For example: `shift=matrix(c(1,1,-1,-1),byrow=TRUE,ncol=2)`

If multiple shifts are supplied, glcm will calculate each texture statistic using all the specified shifts, and return the mean value of the texture for each pixel. To calculate GLCM textures over "all directions" (in the terminology of commonly used remote sensing software), use: `shift=list(c(0,1),c(1,1),c(1,0),c(1,-1)`
This will calculate the average GLCM texture using shifts of 0 degrees, 45 degrees, 90 degrees, and 135 degrees.

Value

A RasterLayer or RasterStack with the requested GLCM texture measures.

References

Lu, D., and M. Batistella. 2005. Exploring TM image texture and its relationships with biomass estimation in Rondônia, Brazilian Amazon. Acta Amazonica 35:249–257.

Gonzalez, R. C. 2008. Digital image processing. 3rd ed. Prentice Hall, Upper Saddle River, N.J, pages 830–836.

Haralick, R. M., K. Shanmugam, and I. Dinstein. 1973. Textural features for image classification. IEEE Transactions on Systems, Man and Cybernetics SMC-3:610–621.

Pratt, W. K. 2007. Digital image processing: PIKS Scientific inside. 4th ed. Wiley-Interscience, Hoboken, N.J pages 540–541, 563–566.

Examples

```
# Calculate GLCM textures on a matrix
d <- matrix(seq(1:25), nrow=5, ncol=5, byrow=TRUE)

# Calculate using default 90 degree shift
glcm(d, statistics=c('variance'))

# Calculate over all directions
glcm(d, shift=list(c(0,1), c(1,1), c(1,0), c(1,-1)),
     statistics=c('variance'))

## Not run:
# Calculate GLCM textures on a raster
require(raster)
# Calculate using default 90 degree shift
textures_shift1 <- glcm(raster(L5TSR_1986, layer=1))
plot(textures_shift1)

# Calculate over all directions
textures_all_dir <- glcm(raster(L5TSR_1986, layer=1),
                         shift=list(c(0,1), c(1,1), c(1,0), c(1,-1)))
plot(textures_all_dir)

## End(Not run)
```

L5TSR_1986 *Landsat 5 Surface Reflectance Image from February 6, 1986 (path 15, row 53)*

Description

Portion of Landsat 5 Surface Reflectance image from the Landsat Climate Data Record archive. This subset of the image includes only bands 1-4.

---

test_raster *Randomly generated 100x100 test image*

Description

Used in testing the output from the GLCM texture statistics C++ code.

Examples

```
# The image was generated with the following code:
require(raster)
set.seed(0)
test_matrix <- matrix(runif(100)*32, nrow=10)
test_raster <- raster(test_matrix, crs='+init=epsg:4326')
test_raster <- cut(test_raster, seq(0, 32))
```

Index