

Package ‘hazer’

May 2, 2018

Title Identifying Foggy and Cloudy Images by Quantifying Hazeness

Version 1.0.1

Date 2018-05-01

Author Bijan Seyednasrollah

Maintainer Bijan Seyednasrollah <bijan.s.nasr@gmail.com>

Description Provides a set of functions to estimate haziness of an image based on RGB bands. It returns a haze factor, varying from 0 to 1, a metric for fogginess and cloudiness. The package also presents additional functions to estimate brightness, darkness and contrast rasters of the RGB image. This package can be used for several applications such as inference of weather quality data and performing environmental studies from interpreting digital images.

Depends R (>= 3.3.0)

Suggests jpeg

License MIT + file LICENSE

Encoding UTF-8

LazyData true

RoxygenNote 6.0.1.9000

URL <https://github.com/bnasr/hazer/>

BugReports <https://github.com/bnasr/hazer/issues>

NeedsCompilation no

Repository CRAN

Date/Publication 2018-05-02 19:43:03 UTC

R topics documented:

getBrightness	2
getContrast	2
getDarkness	3
getHazeFactor	4
plotRGBArray	5
Index	6

getBrightness *The brightness map of an image (0 to 1).*

Description

The brightness map of an image (0 to 1).

Usage

```
getBrightness(rgbaArray)
```

Arguments

rgbaArray RGB array (W x H x 3) where the third dimension contains R, G and B channels, values varying from 0 to 1.

Value

a numeric matrix (W x H) giving the brightness for each pixel of the image.

See Also

[getDarkness](#), [getContrast](#) and [getHazeFactor](#)

Examples

```
library(jpeg)

img <- readJPEG(system.file("img", "Rlogo.jpg", package="jpeg"))

b <- getBrightness(img)

par(mfrow=c(2,1), mar = c(0.5, 1, 0.5, 1))

plotRGBArray(img)
plotRGBArray(b)
```

getContrast *The contrast map of an image (0 to 1).*

Description

The contrast map of an image (0 to 1).

Usage

```
getContrast(rgbaArray)
```

Arguments

rgbArray RGB array (W x H x 3) where the third dimension contains R, G and B channels, values varying from 0 to 1.

Value

a numeric matrix (W x H) giving the contrast for each pixel of the image.

See Also

[getDarkness](#), [getBrightness](#) and [getHazeFactor](#)

Examples

```
library(jpeg)

img <- readJPEG(system.file("img", "Rlogo.jpg", package="jpeg"))

c <- getContrast(img)

par(mfrow=c(2,1), mar = c(0.5, 1, 0.5, 1))

plotRGBArray(img)
plotRGBArray(c)
```

getDarkness	<i>The darkness map of an image (0 to 1).</i>
-------------	---

Description

The darkness map of an image (0 to 1).

Usage

```
getDarkness(rgbArray)
```

Arguments

rgbArray RGB array (W x H x 3) where the third dimension contains R, G and B channels, values varying from 0 to 1.

Value

a numeric matrix (W x H) giving the darkness for each pixel of the image.

See Also

[getContrast](#), [getBrightness](#) and [getHazeFactor](#)

Examples

```
library(jpeg)

img <- readJPEG(system.file("img", "Rlogo.jpg", package="jpeg"))

d <- getDarkness(img)

par(mfrow=c(2,1), mar = c(0.5, 1, 0.5, 1))

plotRGBArray(img)
plotRGBArray(d)
```

getHazeFactor	<i>The haze factor for a given RGB array.</i>
---------------	---

Description

The haze factor for a given RGB array.

Usage

```
getHazeFactor(rgbArray, mu = 5.1, nu = 2.9, sigma = 0.2461)
```

Arguments

rgbArray	RGB array (W x H x 3) where the third dimension contains R, G and B channels, values varying from 0 to 1.
mu	function parameter
nu	function parameter
sigma	function parameter

Value

a numeric value as the haze factor

See Also

[getDarkness](#), [getBrightness](#) and [getContrast](#)

Examples

```
library(jpeg)

img <- readJPEG(system.file("img", "Rlogo.jpg", package="jpeg"))

h <- getHazeFactor(img)
d <- getDarkness(img)
b <- getBrightness(img)
```

```
c <- getContrast(img)

par(mfcol = c(2, 2), mar = c(0.5, 0.5, 0.5, 0.5))

plotRGBArray(img)
mtext(text = 'RGB', side = 3, line = -2, adj = 0.05, font = 2, col = 'red')
mtext(text = paste0('Hazeness: ', signif(h$haze, 2)), side = 1, line = -2, adj = 0.05)
mtext(text = paste0('A0: ', signif(h$A0, 2)), side = 1, line = -1, adj = 0.05)

plotRGBArray(d)
mtext(text = 'Darkness', side = 3, line = -2, adj = 0.05, font = 2, col = 'red')

plotRGBArray(b)
mtext(text = 'Brightness', side = 3, line = -2, adj = 0.05, font = 2, col = 'red')

plotRGBArray(c)
mtext(text = 'Contrast', side = 3, line = -2, adj = 0.05, font = 2, col = 'red')
```

plotRGBArray

Plotting an RGB array on the graphics.

Description

Plotting an RGB array on the graphics.

Usage

```
plotRGBArray(rgbArray, xlim = c(0, 1), ylim = c(0, 1), bty = "o")
```

Arguments

rgbArray	RGB array (W x H x 3) where the third dimension contains R, G and B channels, values varying from 0 to 1.
xlim	range of the x axis.
ylim	range of the y axis.
bty	bty value to pass to the graphics

Value

a rasterImage output plotted on the base R graphics.

See Also

[plotRGBArray](#) wraps the `graphics::rasterImage` function

Examples

```
library(jpeg)
img <- readJPEG(system.file("img", "Rlogo.jpg", package="jpeg"))
plotRGBArray(img)
```

Index

`getBrightness`, [2](#), [3](#), [4](#)

`getContrast`, [2](#), [2](#), [3](#), [4](#)

`getDarkness`, [2](#), [3](#), [3](#), [4](#)

`getHazeFactor`, [2](#), [3](#), [4](#)

`plotRGBArray`, [5](#), [5](#)