Package: visualTest (via r-universe)

September 1, 2024

Title Perform Fuzzy Image Matching

Version 1.0.0

Author Chris Campbell, Gabor Csardi

Maintainer Chris Campbell <ccampbell@mango-solutions.com>

Description Extract and match fingerprints, characteristic signal in
plot files. It can be used for testing packages or other R code
with graphical output. It is possible to set the level of
fuzzyness when comparing the target and the tested images.

License GPL-2

Imports graphics, grDevices, methods, stats, tools

Suggests png, jpeg, bmp, testthat

RoxygenNote 5.0.1

Repository https://predictiveecology.r-universe.dev

RemoteUrl https://github.com/MangoTheCat/visualTest

RemoteRef HEAD

RemoteSha 9b835a707479a9162ca50f108308a5d814bbc923

Contents

---

compareWithFingerprint
Compare a test fingerprint with a known fingerprint

Description

Compare a test fingerprint with a known fingerprint

Usage

```
compareWithFingerprint(test, fingerprint, threshold = 0.001, exact = FALSE)
```

Arguments

test	fingerprint
fingerprint	fingerprint
threshold	single numeric similarity parameter, for the `original` algorithm it defaults to 1e-3. For the `dct` method it is the number bits allowed to be different in the fingerprint (i.e. Hamming distance).
exact	single logical should fingerprints match exactly (default `FALSE`)

Value

single logical

Examples

```
compareWithFingerprint(test = 1:3, fingerprint = 1:3)
compareWithFingerprint(test = 1:3, fingerprint = 1:4)
compareWithFingerprint(test = 1:3, fingerprint = 1:3 + 1e-3)
compareWithFingerprint(test = 1:3, fingerprint = c(1, 2, 3.1))
compareWithFingerprint(
  test = 1:3,
  fingerprint = c(1, 2, 3.1),
  exact = TRUE
)
```

---

getFingerprint   *Get Image Fingerprint*

Description

Get a fingerprint of an image, that does not depend on the fine details of the image. The fingerprint can be used to compare images generated on different machines, platforms, etc. It supports PNG, JPG and BMP images currently.

Usage

```
getFingerprint(file, algorithm = c("dct", "original"))
```

Arguments

`file`	single character naming PNG, JPG or BMP file from which to get fingerprint. It can also be a gzip compressed file with extension `.gz`.
`algorithm`	fingerprint algorithm. Possible values: `original`, `dct`. See details below.

Details

It implements two algorithms. The default algorithm uses a Discrete Cosine Transform (DCT). It first resizes both images into 64x64 size to speed up further calculations. Then it calculates the DCT of both images, takes the top-left 8x8 cells of the DCT, and calculate the difference to the median DCT for both. The result is a 64-bit string represented as a hexadecimal string. The algorithm is similar to and inspired by phash (http://www.phash.org/) and imagehash (https://github.com/jenssegers/imagehash).

The `original` algorithm calculates the Fast Discrete Fourier Transform of both images, column-wise. Then it takes the imaginary parts of the results, sums them up rowwise, and checks when the sums switch sign.

Examples

```
getFingerprint(
  system.file(package = "visualTest", "compare", "stest-00.jpg.gz")
)
```

---

isSimilar   *Is this Graphic a Match to this Fingerprint?*

Description

A function to check similarity between plots. By default, fingerprint comparison is very fuzzy, and the fuzziness can be controlled by argument threshold (see compareWithFingerprint). Fingerprint exact matching can be selected with argument exact. Note that similar looking images may have the same fingerprint (see getFingerprint).

Usage

```
isSimilar(file, fingerprint, threshold = 0.001, exact = FALSE, ...)
```

Arguments

`file`	single character naming PNG or JPG file from which to get fingerprint
`fingerprint`	fingerprint, or filename. Character scalars without a dot in them and numeric vectors are taken as fingerprints.
`threshold`	single numeric similarity parameter (default 1e-3)
`exact`	single logical should fingerprints match exactly (default FALSE). Note that for the `original` algorithm, this is not equivalent to using a zero threshold.
`...`	additional arguments

Examples

```
rdata <- matrix(rnorm(200), ncol = 2)

## To create the images in a temporary directory
tmp <- function(path) file.path(tempdir(), path)

png(tmp("test1.png"))
plot(rdata)
dev.off()

fing1 <- getFingerprint(file = tmp("test1.png"))
rdata[2, 2] <- 0.1
png(tmp("test2.png"))
plot(rdata)
dev.off()

isSimilar(file = tmp("test2.png"), fingerprint = fing1)
isSimilar(file = tmp("test2.png"), fingerprint = fing1, threshold = 0.05)

png(tmp("test3.png"))
plot(rdata, col = 3)
dev.off()

isSimilar(file = tmp("test3.png"), fingerprint = fing1)
isSimilar(file = tmp("test3.png"), fingerprint = fing1, threshold = 0.05)

rdata2 <- matrix(rnorm(200), ncol = 2)
png(tmp("test4.png"))
plot(rdata2)
dev.off()

isSimilar(file = tmp("test4.png"), fingerprint = fing1)
isSimilar(file = tmp("test4.png"), fingerprint = fing1, threshold = 5)

png(tmp("test5.png"))
hist(rdata2)
dev.off()

isSimilar(file = tmp("test5.png"), fingerprint = fing1)
isSimilar(file = tmp("test5.png"), fingerprint = fing1, threshold = 5e6)
```

---

showFingerprint	*Plot some fingerprints against each other This currently only works for fingerprints from the 'original' algorithm.*

Description

Plot some fingerprints against each other This currently only works for fingerprints from the 'original' algorithm.

Usage

```
showFingerprint(..., algorithm = "original")
```

Arguments

...	files to test
algorithm	The algorithm to use, see getFingerprint. Currently only the 'original' algorithm works.

Value

list of fingerprint(s) invisibly.

Examples

```
sf <- system.file(package = "visualTest")
eg <- "VR-616_plot-lm00.jpg.gz"
showFingerprint(
  file.path(sf, "compare", "windows", eg),
  file.path(sf, "compare", "unix", eg)
)
```

Index