Package: colorfindr (via r-universe)

September 14, 2024

Type Package

Title Extract Colors from Windows BMP, JPEG, PNG, TIFF, and SVG Format Images

Version 0.1.4

Description Extracts colors from various image types, returns customized reports and plots treemaps and 3D scatterplots of image compositions. Color palettes can also be created.

License GPL (>= 2)

Encoding UTF-8

LazyData true

BugReports https://github.com/zumbov2/colorfindr/issues

RoxygenNote 6.1.0

Imports purrr, stringr, pixmap, tibble, treemap, rsvg, png, jpeg, tiff, bmp, dplyr, plotly, magrittr, plotwidgets

NeedsCompilation no

Author David Zumbach [aut, cre]

Maintainer David Zumbach <david.zumbach@gfzb.ch>

Repository https://zumbov2.r-universe.dev

RemoteUrl https://github.com/zumbov2/colorfindr

RemoteRef HEAD

RemoteSha 8569b78e1e56bb198a634dc41a51e7febb67ad8c

Contents

get_colors *Extract colors from images.*

Description

get_colors extract colors from Windows BMP, JPEG, PNG, TIFF, and SVG format images.

Usage

```
get_colors(img, exclude_col = NULL, exclude_rad = NULL, top_n = NULL,
  min_share = NULL, get_stats = TRUE)
```

Arguments

img	path or url to image.
exclude_col	vector of colors to be excluded from the analysis. The built-in colors (see colors()) and/or hex color codes can be used.
exclude_rad	numeric vector with blurring of the colors to be excluded. Corresponds to a maximum spherical distance in the RGB color space (all dimensions range from 0 to 255). If is.null, only the exact colors are excluded. If input is of length 1, the same blurring is applied to all elements of exclude_col.
top_n	display the most frequent colors.
min_share	display the colors with a minimum share of all pixels (0-1).
get_stats	if TRUE, absolute and relative frequency of the colors are also included in the response.

Value

If get_stats is set to FALSE a character vector containing the hex color codes is returned. Otherwise, a data.frame (tibble::tibble) is returned with the following columns:

- col_hex hex color code.
- col_freq absolute frequency of the color.
- col_share relative frequency of the color.

Examples

```
# Extract all colors
pic1 <- system.file("extdata", "pic1.png", package = "colorfindr")
get_colors(pic1)

# Extract three most frequent colors
pic2 <- system.file("extdata", "pic2.tif", package = "colorfindr")
get_colors(pic2, top_n = 3)

# Extract colors that fill over 20% of the area
```

```
pic3 <- system.file("extdata", "pic3.jpg", package = "colorfindr")
get_colors(pic3, min_share = 0.2)

# Extract all colors except white
pic4 <- system.file("extdata", "pic4.bmp", package = "colorfindr")
get_colors(pic4, exclude_col = "white")
```

`make_palette` *Create a color palette from an image.*

Description

`make_palette` creates a color palette from colors extracted from Windows BMP, JPEG, PNG, TIFF, and SVG format images with the `get_colors` function.

Usage

```
make_palette(data, n = 10, clust_method = "kmeans",
  extract_method = "hex_freq", show = TRUE)
```

Arguments

`data`	a `data.frame` from a `get_colors` call consisting of the columns `col_hex`, `col_freq`, `col_share`.
`n`	the number of discrete colors to be extracted from the data.
`clust_method`	specifies the method used to cluster the pixels. By default, the colors are clustered by the `k-means` method. Alternatively, a median cut approach `"median_cut"` can be used.
`extract_method`	specifies the process for extracting the colors from the clusters obtained. By default `"hex_freq"`, the most common hex colors per cluster are returned. Alternatively, the cluster-specific`"mean"`, `"median"` or `"mode"` of the RGB values can be used to define the desired number of hex colors.
`show`	by default `"TRUE"`, the generated color palette is displayed.

Value

A character vector with hex color codes, sorted by the weight of the associated clusters.

Examples

```
# Create palette from image
img <- system.file("extdata", "pic6.png", package = "colorfindr")
colors <- get_colors(img)
make_palette(colors)
```

plot_colors *Create treemaps of image color compositions*

Description

plot_colors creates a treemap of colors extracted from Windows BMP, JPEG, PNG, TIFF, and SVG format images with the get_colors function.

Usage

```
plot_colors(data, sort = "color", labels = TRUE)
```

Arguments

data	a data.frame from a get_colors call consisting of the columns col_hex, col_freq, col_share.
sort	specifies the sorting of the treemap rectangles. By default ("color"), the rectangles are sorted by hex color codes, starting in the upper left corner. With ("size") the largest rectangle is placed top left.
labels	by default, rectangles that are sufficiently large are provided with a label. If FALSE, then no labels are displayed.

Examples

```
# Extract all colors
pic1 <- system.file("extdata", "pic1.png", package = "colorfindr")
col <- get_colors(pic1)

# Plot image composition
plot_colors(col)
```

plot_colors_3d *Create interactive 3D scatterplots of image color compositions*

Description

plot_colors_3d calls plotly and creates an interactive 3D scatterplot of colors extracted from Windows BMP, JPEG, PNG, TIFF, and SVG format images with the get_colors function in the RGB color space.

Usage

```
plot_colors_3d(data, sample_size = 5000, marker_size = 2.5,
  color_space = "RGB")
```

Arguments

data	a data.frame from a get_colors call consisting of the columns col_hex, col_freq, col_share.
sample_size	the number of pixels to randomly select.
marker_size	size of marker.
color_space	specifies color space. By default, the colors are displayed in the "RGB" color space (x-axis: red, y-axis: blue, z-axis: green). Alternatively, the color spaces "HSL" (hue, saturation, lightness) and "HSV" (hue, saturation, value) can be used.

Examples

```
# Extract all colors
pic1 <- system.file("extdata", "pic5.png", package = "colorfindr")
col <- get_colors(pic1)

# Plot image composition
plot_colors_3d(col)
```

Index