

Package ‘cropcircles’

July 22, 2025

Type Package

Title Crops an Image to a Circle

Version 0.2.4

URL <https://github.com/doehm/cropcircles>

BugReports <https://github.com/doehm/cropcircles/issues>

Description Images are cropped to a circle with a transparent background. The function takes a vector of images, either local or from a link, and circle crops the image. Paths to the cropped image are returned for plotting with 'ggplot2'. Also includes cropping to a hexagon, heart, parallelogram, and square.

Depends R (>= 3.5.0)

Imports glue, magick, purrr

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.2.3

NeedsCompilation no

Author Daniel Oehm [aut, cre]

Maintainer Daniel Oehm <danieloehm@gmail.com>

Repository CRAN

Date/Publication 2023-12-03 19:40:02 UTC

Contents

add_border	2
crop_circle	2
cut_circle	5
cut_heart	5
cut_hex	6
cut_parallelogram	6
cut_square	7

Index	8
--------------	----------

add_border	<i>Add border helper</i>
------------	--------------------------

Description

Add border helper

Usage

```
add_border(x, geom, border_size, border_colour, bg_fill, orig)
```

Arguments

x	magick image
geom	Geometric shape e.g. circle, hex, heart.
border_size	Border size in pixels.
border_colour	Border colour
bg_fill	Background fill
orig	List of original dimensions e.g. 'list(wd = 100, ht = 200)'

Value

Magick image

crop_circle	<i>Cropping functions</i>
-------------	---------------------------

Description

Reads in an image and crops to the specified geometry with a transparent background. If a new path is given it will save the cropped images to the new location. If no path is given it will save to a temporary location which will be cleared when the session is closed

Usage

```
crop_circle(
  images,
  to = NULL,
  border_size = NULL,
  border_colour = "black",
  bg_fill = NULL,
  just = "center"
)
```

```
crop_square(  
    images,  
    to = NULL,  
    border_size = NULL,  
    border_colour = "black",  
    bg_fill = NULL,  
    just = "center"  
)
```

```
crop_hex(  
    images,  
    to = NULL,  
    border_size = NULL,  
    border_colour = "black",  
    bg_fill = NULL,  
    just = "center"  
)
```

```
crop_heart(  
    images,  
    to = NULL,  
    border_size = NULL,  
    border_colour = "black",  
    bg_fill = NULL,  
    just = "center"  
)
```

```
crop_parallelogram(  
    images,  
    to = NULL,  
    border_size = NULL,  
    border_colour = "black",  
    bg_fill = NULL,  
    just = "center"  
)
```

```
circle_crop(  
    images,  
    to = NULL,  
    border_size = NULL,  
    border_colour = "black",  
    bg_fill = NULL,  
    just = "center"  
)
```

```
hex_crop(  
    images,  
    to = NULL,
```

```

border_size = NULL,
border_colour = "black",
bg_fill = NULL,
just = "center"
)

```

Arguments

images	Vector of image paths, either local or urls. If urls the images will be downloaded first.
to	Path to new location
border_size	Border size in pixels.
border_colour	Border colour.
bg_fill	Background fill. Allows a different colour for the background and a different colour for the border.
just	Where to justify the image prior to cropping. Accepted values: 'left', 'right', 'top', 'bottom'

Value

Path to cropped images

Note

The naming convention is now 'crop_*'. The old functions 'circle_crop' and 'hex_crop' still work but you are encouraged to use the new functions 'crop_circle' and 'crop_hex'.

Examples

```

library(cropcircles)
library(magick)

img_path <- file.path(system.file(package = "cropcircles"), "images", "walter-jesse.png")
img_cropped <- crop_circle(img_path, border_size = 6)
image_read(img_cropped)

# justification example

# center (default)
image_read(crop_circle(img_path, border_size = 6))

# left
image_read(crop_circle(img_path, border_size = 6, just = "left"))

# right
image_read(crop_circle(img_path, border_size = 6, just = "right"))

```

cut_circle	<i>Circle crop helper</i>
------------	---------------------------

Description

Circle crop helper

Usage

```
cut_circle(x, just = "center")
```

Arguments

x	Magick images
just	Where to justify the image prior to cropping

Value

Magick image

cut_heart	<i>heart crop helper</i>
-----------	--------------------------

Description

heart crop helper

Usage

```
cut_heart(x, just = "center")
```

Arguments

x	Magick image
just	Where to justify the image prior to cropping

Value

Magick images

cut_hex	<i>Hex crop helper</i>
---------	------------------------

Description

Hex crop helper

Usage

```
cut_hex(x, just = "center")
```

Arguments

- | | |
|------|--|
| x | Magick image |
| just | Where to justify the image prior to cropping |

Value

Magick image

cut_parallelogram	<i>Parallelogram crop helper</i>
-------------------	----------------------------------

Description

Parallelogram crop helper

Usage

```
cut_parallelogram(x, just = "center")
```

Arguments

- | | |
|------|--|
| x | Magick image |
| just | Where to justify the image prior to cropping |

Value

Magick image

cut_square	<i>Square crop helper</i>
------------	---------------------------

Description

Square crop helper

Usage

```
cut_square(x, just = "center")
```

Arguments

- | | |
|------|-------------------------------------|
| x | Magick images |
| just | Where to justify the image prior to |

Value

Magick image

Index

`add_border`, [2](#)

`circle_crop(crop_circle)`, [2](#)

`crop_circle`, [2](#)

`crop_heart(crop_circle)`, [2](#)

`crop_hex(crop_circle)`, [2](#)

`crop_parallelogram(crop_circle)`, [2](#)

`crop_square(crop_circle)`, [2](#)

`cut_circle`, [5](#)

`cut_heart`, [5](#)

`cut_hex`, [6](#)

`cut_parallelogram`, [6](#)

`cut_square`, [7](#)

`hex_crop(crop_circle)`, [2](#)