

# Package: coiR (via r-universe)

October 24, 2026

**Type** Package

**Title** Analysis of Canopy Openness Index from Fisheye Images

**Version** 0.1.0

**Description** Tools to estimate the Canopy Openness Index (COI) from hemispherical (fisheye) canopy photographs. The package imports canopy images, crops them to the circular area delimited by the fisheye lens, binarizes them to separate sky from vegetation pixels, and computes the index as the proportion of sky pixels in the cropped image. Canopy openness is a measure related to canopy closure, canopy cover and understorey illumination, as discussed in Jennings, Brown and Sheil (1999) ``Assessing forest canopies and understorey illumination: canopy closure, canopy cover and other measures" <doi:10.1093/forestry/72.1.59>. Example canopy images are included so users can test the workflow.

**URL** <https://github.com/Edbbioeco/coiR>

**BugReports** <https://github.com/Edbbioeco/coiR/issues>

**License** MIT + file LICENSE

**Encoding** UTF-8

**Depends** R (>= 4.1)

**Imports** ggplot2, dplyr, terra, tidyterra, rlang

**Suggests** testthat (>= 3.0.0), purrr

**Config/testthat/edition** 3

**Config/roxygen2/version** 8.1.0

**NeedsCompilation** no

**Author** Edson Nilton de Moura Silva-Júnior [aut, cre]

**Maintainer** Edson Nilton de Moura Silva-Júnior  
<edsonbbiologia@gmail.com>

**Repository** <https://cran.r-universe.dev>

**Date/Publication** 2026-10-24 11:17:38 UTC

**RemoteUrl** <https://github.com/cran/coiR>  
**RemoteRef** HEAD  
**RemoteSha** 6decc5906fd81f7f4b6ffc038f3f4bb93fd40f0f

**Contents**

|                         |          |
|-------------------------|----------|
| coir_binarize . . . . . | 2        |
| coir_crop . . . . .     | 3        |
| coir_index . . . . .    | 4        |
| data_coir . . . . .     | 6        |
| <b>Index</b>            | <b>8</b> |

---

|               |                                               |
|---------------|-----------------------------------------------|
| coir_binarize | <i>Binarize cropped square fisheye images</i> |
|---------------|-----------------------------------------------|

---

**Description**

To analyze Canopy Opennes Index (COI), we need to difference what is interpreted in the image as canopy and what is interpred as sky. As our images are usely RGB (colored) images, a first step is to decompond our images into a black-and-white image, where we get a gradient from black (usely, value = 0) to white (usely, value = 1). Next, to binarize our image consists only to set a threshold (usely, >= 0.5, but check details), where values < threshold are interpreted as canopy and values >= threshold are interpreted as sky.

**Usage**

```
coir_binarize(data, threshold = 0.5, plot = TRUE)
```

**Arguments**

- data            A canopy image, as a SpatRast class objecr, from terra package.
- threshold      Numeric. A value to definy what value from is considered sky.
- plot           default is TRUE. If TRUE, function plot circular cropped image.

**Details**

The threshold value is based on bridghtness index in pluvial forests, but that must be setted by image bridghtness index.

**Value**

Um objeto da classe ‘SpatRaster’ (pacote ‘terra’) binarizado, em que cada pixel é classificado como céu ou vegetação.

**Examples**

```
# Load packages

library(coiR)

library(purrr)

# Importing data

images <- coiR::data_coiR()

image_single <- images[[1]]

binarized_image <- image_single |>
  coiR::coir_crop() |>
  coiR::coir_binarize()

binarized_image

purrr::map(images, \(caminho){

  binarized <- caminho |>
    coiR::coir_crop() |>
    coiR::coir_binarize()

  print(binarized)

})
```

coir\_crop

*Crop square fisheye images into circle images***Description**

To analyze Canopy Openness Index (COI), we shot images using a fisheye lens. Nevertheless that usability, fisheye images have null spaces, due image distortion. To avoid to overestimate canopy closure, we need to previously crop the images.

**Usage**

```
coir_crop(data, plot = TRUE)
```

**Arguments**

|      |                                                                        |
|------|------------------------------------------------------------------------|
| data | A canopy image, as a SpatRast class object, from <b>terra</b> package. |
| plot | Default is TRUE. If TRUE, function plot circular cropped image.        |

**Details**

That function must to be used only for images were not cropped. Wheter your data are images previously circular cropped, that function are not required to calculate COI.

**Value**

Um objeto da classe ‘SpatRaster’ (pacote ‘terra’) com a imagem recortada na área circular da lente fisheye.

**Examples**

```
# Load packages

library(coiR)

library(purrr)

# Importing data

images <- coiR::data_coir()

# Isolating a one single image

image_single <- images[[1]]

# Crop images

cropped_image <- coiR::coir_crop(image_single)

cropped_image

# Crop multiple images

purrr::map(images, \(caminho){

  cropped <- caminho |>
    coiR::coir_crop()

  print(cropped)

})
```

## Description

Canopy Openness Index (COI) is a value to define how much open is a canopy. This values ranges from 0 (full cole canopy) to 1 (full open canopy). to that function, we use a cropped binarized canopy image (see details).

## Usage

```
coir_index(data, round = 2)
```

## Arguments

|       |                                                                                               |
|-------|-----------------------------------------------------------------------------------------------|
| data  | A cropped binarized canopy image, as a SpatRast class objecr, from <b>terra</b> pack-<br>age. |
| round | Numeric. Default is 2. Round output decimal place count.                                      |

## Details

It is possible to use previous image, but raw canopy images can be used through using `coiR::coir_crop()` and `coiR::coir_binarize()` functions.

## Value

Um valor numérico ('numeric' de comprimento 1) com o Índice de Abertura do Dossel (COI), calculado como a proporção de pixels de céu na imagem binarizada. Valores próximos de 0 indicam dossel fechado e valores próximos de 1 indicam dossel aberto.

## Examples

```
# Load packages

library(coiR)

library(purrr)

# Importing data

images <- coiR::data_coir()

# Isolating a one single image

image_single <- images[[1]]

# Get image index

image_index <- image_single |>
  coiR::coir_crop() |>
  coiR::coir_binarize() |>
  coiR::coir_index()

image_index
```

```
# Binarize multiple images

purrr::map(images, \(caminho){

  index <- caminho |>
  coiR::coir_crop() |>
  coiR::coir_binarize() |>
  coiR::coir_index()

  print(index)

})
```

---

data\_coir

*Data for examples*


---

## Description

Simulated data to use in coiR package functions. Canopy images as a SpatRast class object, from **terra** package (see details).

## Usage

```
data_coir()
```

## Format

A set list with 4 images, with different canopy openness:

## Details

images are setted by a list of 4 rasters (SpatRast class). Pay attention to exemple to know how procede with a single isolated image and a list of images.

## Value

Um vetor de caracteres ('character') com os caminhos completos das imagens de dossel de exemplo (formato PNG) instaladas com o pacote. Esses caminhos podem ser usados como entrada das demais funções, como 'coir\_crop()'.

## Source

Images manually shotten by author using a fisheye lens, not generated by AI.

**Examples**

```
# Load packages

library(coiR)

# Importing data

images <- coiR::data_coir()

# Visualizing

purrr::map(images, \(caminho){

  imagem <- caminho |> terra::rast()

  ggplt <- ggplot2::ggplot() +
    tidyterra::geom_spatraster_rgb(data = imagem)

  print(ggplt)

})
```

# Index

coir\_binarize, [2](#)

coir\_crop, [3](#)

coir\_index, [4](#)

data\_coir, [6](#)