Package: image.libfacedetection (via r-universe)

October 25, 2026

Type Package

Title Convolutional Neural Network for Face Detection

Description An open source library for face detection in images. Provides a pretrained convolutional neural network based on <https://github.com/ShiqiYu/libfacedetection> which can be used to detect faces which have size greater than 10x10 pixels.

Maintainer Jan Wijffels <jwijffels@bnosac.be>

License BSD_3_clause + file LICENSE

Version 0.1.1

URL https://github.com/bnosac/image

Imports Rcpp (>= 0.12.8), graphics

LinkingTo Rcpp

Suggests magick

RoxygenNote 7.1.2

NeedsCompilation yes

Author Jan Wijffels [aut, cre, cph] (R wrapper), BNOSAC [cph] (R wrapper), Shiqi Yu [ctb, cph] (libfacedetection C++ code)

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

Date/Publication 2025-11-26 22:30:02 UTC

RemoteUrl https://github.com/cran/image.libfacedetection

RemoteRef HEAD

RemoteSha fca214594e108d94bdd7bc817b792780d8beb407

Contents

image_detect_faces 2
plot.libfacedetection 3

Index **5**

image_detect_faces *Detect faces in images using the libfacedetection CNN*

Description

Detect faces in images using using a convolutional neural network available from https://github.com/ShiqiYu/libfacedetection. The function can be used to detect faces of minimal size 10x10 pixels.

Usage

```
image_detect_faces(x)
```

Arguments

x	an object of class magick-image with rgb colors. Or an rgb integer array with pixel values in the 0-255 range.

Value

A list with elements nr and detections.
Element nr indicates the number of faces found.
The data frame detections indicates the locations of these. This data.frame has columns x, y, width and height as well as a columns called confidence. The values of x and y are the top left of the start of the box. This data frame also has the x and y locations of 5 face landmarks (eyes, nose and mouth ends).

Examples

```
library(magick)
path <- system.file(package="image.libfacedetection", "images", "handshake.jpg")
x <- image_read(path)
x
faces <- image_detect_faces(x)
faces
plot(faces, x, border = "red", lwd = 7, col = "white", landmarks = TRUE)


##
## You can also directly pass on the RGB array in BGR format
## without the need of having magick
##
tensor <- image_data(x, channels = "rgb")
tensor <- as.integer(tensor)
faces  <- image_detect_faces(tensor)
str(faces)
plot(faces, x)
```

`plot.libfacedetection` *Plot detected faces*

Description

Plot functionality for bounding boxes detected with `image_detect_faces`

Usage

```
## S3 method for class 'libfacedetection'
plot(
  x,
  image,
  border = "red",
  lwd = 5,
  only_box = FALSE,
  col = "red",
  cex = 2,
  landmarks = FALSE,
  col_landmarks = "black",
  cex_landmarks = 1,
  pch_landmarks = 20,
  ...
)
```

Arguments

`x`	object of class `libfacedetection` as returned by `image_detect_faces`
`image`	object of class `magick-image` which was used to construct `x`
`border`	color of the border of the box. Defaults to red. Passed on to `rect`
`lwd`	line width of the border of the box. Defaults to 5. Passed on to `rect`
`only_box`	logical indicating to draw only the box and not the text on top of it. Defaults to FALSE.
`col`	color of the text on the box. Defaults to red. Passed on to `text`
`cex`	character expension factor of the text on the box. Defaults to 2. Passed on to `text`
`landmarks`	logical indicating to plot the landmarks as points. Defaults to FALSE.
`col_landmarks`	color of the point of the landmarks. Defaults to black.
`cex_landmarks`	cex of the point of the landmarks. Defaults to 1.
`pch_landmarks`	pch of the point of the landmarks. Defaults to 20.
`...`	other parameters passed on to `rect`

Value

an object of class `magick-image`

Examples

```
library(magick)
path <- system.file(package="image.libfacedetection", "images", "handshake.jpg")
x <- image_read(path)
x
faces <- image_detect_faces(x)
faces
plot(faces, x, border = "red", lwd = 7, col = "white")
plot(faces, x, border = "red", lwd = 7, col = "white", landmarks = TRUE,
     col_landmarks = "purple", cex_landmarks = 2, pch_landmarks = 4)

## show one detected face
face <- head(faces$detections, 1)
image_crop(x, geometry_area(x = face$x, y = face$y,
                            width = face$width, height = face$height))
## show all detected faces
boxcontent <- lapply(seq_len(faces$nr), FUN=function(i){
  face <- faces$detections[i, ]
  image_crop(x, geometry_area(x = face$x, y = face$y,
                              width = face$width, height = face$height))
})
boxcontent <- do.call(c, boxcontent)
boxcontent
```

Index

image_detect_faces, 2, *3*

plot.libfacedetection, 3

rect, *3*

text, *3*