Package ‘image.binarization’

October 13, 2022

Type Package
Title Binarize Images for Enhancing Optical Character Recognition
Version 0.1.3
Maintainer Jan Wijffels <jwijffels@bnosac.be>
Description Improve optical character recognition by binarizing images. The package focuses primarily on local adaptive thresholding algorithms.
In English, this means that it has the ability to turn a color or gray scale image into a black and white image. This is particularly useful as a preprocessing step for optical character recognition or handwritten text recognition.
License MPL-2.0
URL https://github.com/DIGI-VUB/image.binarization
Encoding UTF-8
Depends R (>= 4.0.0)
Imports Rcpp, magick, grDevices
LinkingTo Rcpp
RoxygenNote 7.1.2
SystemRequirements C++17
NeedsCompilation yes
Author Jan Wijffels [aut, cre, cph] (R wrapper),
Vrije Universiteit Brussel - DIGI: Brussels Platform for Digital Humanities [cph] (R wrapper),
Brandon M. Petty [ctb, cph] (Files in src/Doxa)
Repository CRAN
Date/Publication 2022-08-17 10:50:09 UTC

R topics documented:

image_binarization ...................................................... 2

Index 4
Binarize Images For Enhancing Optical Character Recognition

Description

Binarize images in order to further process it for Optical Character Recognition (OCR) or Handwritten Text Recognition (HTR) purposes


Usage

image_binarization(x, type, opts = list())

Arguments

x an image of class 'magick-image'. In grayscale. E.g. a PGM file. If not provided in grayscale, will extract the gray channel.

type a character string with the type of binarization to use. Either 'otsu', 'bernsen', 'niblack', 'sauvola', 'wolf', 'nick', 'gatos', 'su', 'trsingh', 'bataineh', 'wan' or 'isauvola'

opts a list of options to pass on to the algorithm. See the details and the examples.

Details

Options which can be bassed on to the binarization routines, with the defaults between brackets

- otsu: none
- bernsen: window(75L), k(0.2), threshold(100L), contrast-limit(25L)
image_binarization

- niblack: window(75L), k(0.2)
- sauvola: window(75L), k(0.2)
- wolf: window(75L), k(0.2)
- nick: window(75L), k(-0.2)
- gatos: window(75L), k(0.2), glyph(60L)
- su: window(75L), minN(75L)
- trsingh: window(75L), k(0.2)
- bataineh: none
- wan: window(75L), k(0.2)
- isauvola: window(75L), k(0.2)

Note that it is important that you provide the window / threshold / contrast-limit, minN, glyph argument as integers (e.g. as in 75L) and the other parameters as numerics.

Value

a binarized image of class magick-image as handled by the magick R package

Examples

library(magick)
f <- system.file("extdata", "doxa-example.png", package = "image.binarization")
img <- image_read(f)
img <- image_convert(img, format = "PGM", colorspace = "Gray")

binary <- image_binarization(img, type = "otsu")
binary
binary <- image_binarization(img, type = "bernsen",
opts = list(window = 50L, k = 0.2, threshold = 50L))
binary
binary <- image_binarization(img, type = "niblack", opts = list(window = 75L, k = 0.2))
binary
binary <- image_binarization(img, type = "sauvola")
binary
binary <- image_binarization(img, type = "wolf")
binary
binary <- image_binarization(img, type = "nick", opts = list(window = 75L, k = -0.2))
binary
binary <- image_binarization(img, type = "gatos", opts = list(window = 75L, k = 0.2, glyph = 50L))
binary
binary <- image_binarization(img, type = "su", opts = list(window = 20L))
binary
binary <- image_binarization(img, type = "trsingh")
binary
binary <- image_binarization(img, type = "bataineh")
binary
binary <- image_binarization(img, type = "wan")
binary
binary <- image_binarization(img, type = "isauvola", opts = list(window = 75L, k = 0.2))
binary
Index

image_binarization, 2