Package 'CropDetectR'

October 12, 2022

Title Crop Row Detector

Version 0.0.1

Description A helpful tool for the identification of crop rows. Methods of this package include: Excess Green color scale <https://www.researchgate.net/publication/270613992_Color_Indices_for_Weed_Identification_Under_Various_Soil_Residue_and_Lighting_Conditions>, Otsu Thresholding <https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=4310076>, and Morphology <https://en.wikipedia.org/wiki/Mathematical_morphology>.

Depends R (>= 3.1.0)

License GPL-3

BugReports https://github.com/niconaut/CropDetectR/issues

Encoding UTF-8

LazyData true

Imports dplyr, imager, reshape2, EBImage, stats

Suggests knitr, ggplot2, rmarkdown, testthat

VignetteBuilder knitr

RoxygenNote 6.1.1

NeedsCompilation no

Author Nicolaas VanSteenbergen [aut, cre]

Maintainer Nicolaas VanSteenbergen <nvansteenbergen@unomaha.edu>

Repository CRAN

Date/Publication 2019-09-20 08:10:02 UTC

R topics documented:

`best_rotation` *Find the best rotated image for crop row detection*

Description

Analyzes the list of rotated images of crop rows and finds the best angle of rotation for row detection.

Usage

```
best_rotation(picture_list, ratio, intensity)
```

Arguments

`picture_list`	One or more images created from rotating the same image.
`ratio`	Any number, typically (0-1) that will be the ratio needed to determine a true crop row.
`intensity`	The amount of smoothing of the image.

Details

This takes in a list of images and looks at the average of each column in the form of an arry. Ideally the image is black and white, with crops being white, so the range of numbers in the array 0,1.

The function then smooths out the array using `smoothing` so the local minima and maximas using (`localMaxima` and `localMaxima`) are more pronounced, then creates two vectors of the local min/max of each image. Then the ratio between neighbor local minima and maxima are calculated and compared to a threshold given by the user. If the found ratio is larger than the threshold the ratio is counted.

Value

The index of the image with the most *good ratios* (ratios that exceed the given threshold).

Examples

```
best_image <- best_rotation(picture_list, 0.5, 0.25)
```

blobify — *Denoise a black and white image to only core features*

Description

This takes a black and white image, preferrably the image produced by an otsu transformation, and gets rid of all extrenuous features. Many long skinny features and random clumps of features will be erased, leaving only the larger core features of an image.

Usage

```
blobify(image, size)
```

Arguments

image	An image.
size	An odd number, the starting size of the kernel for morphology, minimum of 3 starting size.

Details

Using morphology functions https://en.wikipedia.org/wiki/Mathematical_morphology to get rid of unwanted noise and features. With this package the features to eliminate will mostly be random weeds and bushy leaves. The specific functions being use are opening https://en.wikipedia.org/wiki/Opening_(morphology) and closing https://en.wikipedia.org/wiki/Closing_(morphology). The first step of the function is opening to start to get rid of smaller features. It then closes the holes created by opening the image and repeats this process several times with bigger and bigger kernels. This allows more important featrures to remain while insignificant ones are erased. There is a minimum starting size of 2, because when doing the first closing process it needs to be at least a 1 pixel kernel.

Examples

```
final_blob <- blobify(image, size)
```

crop_lines — *Find crop lines of an image*

Description

Finds the crop row lines of an image. It uses other functions in the package such as: smoothing, localMaxima, and localMinima.

Usage

```
crop_lines(picture_list, final_ratio, best_image, intensity)
```

Arguments

picture_list	One or more images created from rotating the same image.
final_ratio	Any number, typically (0-1) that will be the ratio needed to determine a true crop row.
best_image	The output of the best_rotations function.
intensity	The amount of smoothing of the image.

Details

This function takes in: one or more images, a ratio that will become the threshold of what qualifies as a crop row, the index from best_rotation of what the best image was, and the intensity of smoothening from smoothing. It then uses the ratio as the threshold for acceptable crop rows and makes a list of the x-axis values.

Value

A vector of x-axis values for a given image.

Examples

```
crop_positions <- crop_lines(picture_list, 0.5, best_image, 0.25)
```

---

crop_row_finder *Maps out the crop rows of the image (of a maize field)*

Description

Finds the x coordinates of the crop rows in the image.

Usage

```
crop_row_finder(picture_list, ratio, final_ratio, intensity)
```

Arguments

picture_list	The list of rotated images originally from a single image.
ratio	The first strict ratio used to identify which rotation has the most vertical crop rows.
final_ratio	The less-strict ratio used on the best image after rotations to capture smaller potential crop rows.
intensity	The amount of smoothing of the image.

Details

This function is a combination of two previous functions `best_rotation` and `crop_lines` to find the best rotation and then map out the x coordinates of the crop rows in the image.

Value

a list of x coordinates for the image on which the function was applied.

Examples

```
crop_rows <- crop_row_finder(picture_list, 0.5, 0.05, 0.25)
```

---

localMaxima *Finds local maxima of a vector*

Description

Takes in a vector and finds the local maximas. (Credit user Tommy https://stackoverflow.com/questions/6836409/finding-local-maxima-and-minima).

Usage

```
localMaxima(x)
```

Arguments

x	A 1D array or vector

Value

A list of local maximas in the vector

Examples

```
maximas <- localMaxima(x)
```

localMinima *Finds local minima of a vector*

Description

Takes in a vector and finds the local minimas. (Credit user Tommy https://stackoverflow.com/questions/6836409/finding-local-maxima-and-minima).

Usage

```
localMinima(x)
```

Arguments

x	Any vector

Value

A list of local minimas in the vector

Examples

```
minimas <- localMinima(x)
```

make_bw *Changes a grayscale image to black and white*

Description

Takes in a grayscale image and finds the best threshold for binarization of the image.

Usage

```
make_bw(image)
```

Arguments

image	The image generated after the ExG transformation in the EBImage format.

Details

This function uses the https://github.com/aoles/EBImage/blob/master/R/otsu.R otsu function from EBImage to make the grayscale image into a binary black and white image. How the otsu transformation works and chooses the threshold can be understood more clearly at http://www.labbookpages.co.uk/software/imgProc/otsuThreshold.html.

Value

A binary image

Examples

```
BW_Image <- make_bw(image)
```

`make_ExG` *Converts a color image into a grayscale image using ExG methodology.*

Description

Uses the Excessive Green (ExG) methodology to create a grayscale image of crop rows. The image is first broken down into a dataframe and each pixel is tested then put onto a varying intensity according to the ExG.

Usage

```
make_ExG(color_image)
```

Arguments

`color_image`	The image to have ExG applied to it

Details

The function first reads the image as a data frame with x and y columns for the pixel postion. Reading the data frame as "wide = 'c'" also creates a column for red, green, and blue intensity. Each color column is then normalized and the normal values of the colors are put into the ExG equation, creating a new ExG column for each pixel. The values of the ExG are then treated as a 1D array and transformed into a format for the EBImage package commands.

Value

An image formatted for EBImage commands.

Examples

```
grayscale <- make_ExG(color_image)
```

`rotations` *Rotates an image by x degrees*

Description

Takes an image and rotates it by a number of degrees choosen by the user.

Usage

```
rotations(picture, degrees)
```

Arguments

`picture`	The image to be rotated.
`degrees`	The degrees of rotation until 180 (30 = every 30 degrees).

Details

The function takes in an image and rotates it by a number of degrees chosen by the user. It will keep rotating until it has reached the 360 degree limit and save the images into a list.

Value

A list of pictures that have been rotated by x degrees each.

Examples

```
picture_list <- rotations(picture, 45)
```

`smoothing` *Smooth the average of binary picture column values*

Description

Reads in an image and takes the average column value then smooths the array for more defined local maximas and minimas.

Usage

```
smoothing(picture, intensity)
```

Arguments

`picture`	The binary image.
`intensity`	The intensity of smoothening of the vector.

Details

The function reads in an image as a data frame then takes the mean of each column within the picture. Since the image is binary the mean will have a value between [0,1], with one being all white. It then uses https://www.rdocumentation.org/packages/stats/versions/3.6.1/topics/smooth.spline smooth.spline to get rid of jagged portions for more defined local maximas and minimas.

Value

A smoothed vector.

Examples

```
smoothed_vector <- smoothing(picture, intensity)
```

Index