Bern img utils

Generated by Doxygen 1.8.11

Contents

Chapter 1

bern_img_utils

Useful helper functions of image processing and machine learning

Libraries

Libraries contained in the repository

dataframe.py

Useful functions to manipulate pandas dataframe

functions.py

Random useful functions

images.py

Image processing functions

masks.py

Functions to modify binary masks

LikelihoodGenerator.py

Class to help to generate likelihood masks of regions of interest in images.

Build your likelihood by marking with the mouse the contours of the roi region.

Keyboard commands:

key a: Submit region selected.
key b: discard last mouse click

Example of use:

```
1 image = "RGB image matrix"
2 path_output_file = "Path to save the likelihood"
3 LikelihoodGenerator(image).build_your_mask(path_output_file)
```

Methods

build_your_mask(path)

Call this function to init the process of finger_regions selection.

In the process:

Press A (shift + a) to end selection

Press b to cancel the last selection

When process ends the method return the mask build. This mask is the region Inside the selected zone..

plots.py

Functions to help plotting images.

Chapter 2

Namespace Index

2.1 Namespace List

Here is a list of all namespaces with brief descriptions:

Chapter 3

Class Index

3.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 4

File Index

4.1 File List

Here is a list of all files with brief descriptions:

Chapter 5

Namespace Documentation

5.1 dataframe Namespace Reference

Functions

- def df_attr_split (df, attr, value, columns)
- def df_filter (df, filters)

5.1.1 Function Documentation

5.1.1.1 def dataframe.df_attr_split (*df, attr, value, columns*)

```
Separe a dataframe by an atribute value
:param df: Dataframe
:param attr: String name of the colums/attribute to split by
:param value: Any value of attr to select
:param columns: String[] list of columns to get of the dataframe
:return:
```

5.1.1.2 def dataframe.df_filter (*df, filters*)

```
Filter DataFrame Content by a list
:param df: DataFrame witch want to filter
:param filters: dict select the items to remove to the dataframe
        key => column, value => array of values of column of rows to discard
:return:
```

5.2 functions Namespace Reference

Functions

- def potencia (input, min_n_items=1, max_n_items=None)
- def check_dir (path)

5.2.1 Function Documentation

5.2.1.1 def functions.check_dir (*path*)

```
Check if the specified path exists into file system. If not exists, it is created
:param path: String
:return:
```

5.2.1.2 def functions.potencia (*input*, *min_n_items* = 1, *max_n_items* = None)

```
Calcula y devuelve el conjunto potencia del
   conjunto input.
```

5.3 images Namespace Reference

Functions

- def binary2RGB (image)
- def grays2binary (image_grays)

5.3.1 Function Documentation

5.3.1.1 def images.binary2RGB (*image*)

```
Convert a binary image to RGB, black & white image
O values to BLACK others to white
:param image: binary image
:return: binary image coverted to RGB, true values (or 1) to BLACK and other to WHITE
```

5.3.1.2 def images.grays2binary (*image_grays*)

```
Convert a grayscale image to a binary one
:param image_grays: one chanel image
:return:
```

5.4 LikelihoodGenerator Namespace Reference

Classes

- class LikelihoodGenerator

5.5 masks Namespace Reference

Functions

- def mask_evaluation (mask, likelihood)
- def mask_sklearn_evaluation (mask, likelihood)
- def masks_coincidence (mask1, mask2, priority="big_mask")
- def mask_onto_mask (mask1, mask2, perc=0.9)
- def mask_2RGB (mask)
- def mask_fill_holes (mask)
- def mask_from_RGB_file (file_mask)
- def mask_bounding_circle (mask)
- def mask_delete_contour_in (mask, region)
- def mask_build_circular (image, circle)
- def mask_build_circular_boolean (image, circle)
- def mask_biggest_connected_component (mask)

5.5.1 Function Documentation

5.5.1.1 def masks.mask_2RGB (*mask*)

```
Convert a binary image to RGB, black & white image
:param mask: binary image
:return: binary image coverted to RGB, true values (or 1) to BLACK and other to WHITE
```

5.5.1.2 def masks.mask_biggest_connected_component (*mask*)

```
Get the biggest connected component of a mask
:param mask:
:return:
```

5.5.1.3 def masks.mask_bounding_circle (*mask*)

```
Get the minium enclosing circle of a given mask
:param mask:
:return: (x, y), r => integers
```

5.5.1.4 def masks.mask_build_circular (*image, circle*)

```
Build a circular mask onto the image
:param image: 3 channels image
:param circle: (int,int, int)
    tuple with the circle to find, two first values are circle coordinates (x, y), the thirth is the radius
:return:
```

5.5.1.5 def masks.mask_build_circular_boolean (*image, circle*)

```
build a circular binary onto the image
:param image: image
:param circle: (int,int, int)
    tuple with the circle to find, two first values are circle coordinates (x, y), the thirth is the radius
:return:
```

5.5.1.6 def masks.mask_delete_contour_in (*mask, region*)

```
:param mask:
:param region:
:return:
```

5.5.1.7 def masks.mask_evaluation (*mask, likelihood*)

```
Get the evaluation metrics of a given mask and his likelihood
:param mask: Binary image => Mask to evaluate
:param likelihood: Binary image => Correct mask
:raise Incorrect likelihood
:return: dict{
    "FN"
    "TP"
    "FP"
    "Recall"
    "Precision"
    "F1"
    "cohen_kappa"
    "accuracy"
}
```

5.5.1.8 def masks.mask_fill_holes (*mask*)

```
Fill all the empty pixels overwhelmed by true pixels
:param mask: 0, 255 mask
:return: O values inside 255 values filled with 255
```

5.5.1.9 def masks.mask_from_RGB_file (*file_mask*)

```
Build a mask from a BLACK & white RGB image from file
:param file_mask: path of the file to build
:return: uint8 image
```

5.5.1.10 def masks.mask_onto_mask (*mask1, mask2, perc* = 0.9)

```
Given two masks of the same shape, check if them are one onto the other.

:param mask1:
:param mask2:
:param perc: percentage of coincidence pixels that must have the two masks
                to consider that them are one onto the other.
:return: bool
```

5.5.1.11 def masks.mask_sklearn_evaluation (*mask, likelihood*)

```
Same than mask_evaluation, but using sklearn library for compute values
:param mask:
:param likelihood:
:return:
```

5.5.1.12 def masks.masks_coincidence (*mask1, mask2, priority* = "big_mask")

```
Get the porcentage of coincidence between two masks of the same shape
:param priority:
:param mask1:
:param mask2:
:return:
```

5.6 plots Namespace Reference

Functions

- def multiplot (images, filename=None, nrows=2, colorbar=False)
- def plots_correlation_matrix (df, labels=None, absolute=False)
- def plots_segmentation (img, labels)
- def plots_raw_data (df, columns, colors="b")
- def biplot (pca, dat)

5.6.1 Function Documentation

5.6.1.1 def plots.biplot (*pca, dat*)

```
Plot a data biplot on the screen
:param dat: DataFrame data to plot
:return:
```

5.6.1.2 def plots.multiplot (*images, filename* = None, *nrows* = 2, *colorbar* = False)

```
Help to plot a multiple image figure
:param images: list of dictionaries with structure:
    {
    "img": image to plot,
    "title": name of the plot
    }
:param filename: String|None
                    Path file Where save the figure,
                    None to plot in a window
:param colorbar: bool
                    true to plot a colorbar near each image
:return:
```

5.6.1.3 def plots.plots_correlation_matrix (*df, labels* = None, *absolute* = False)

```
Get correlation matrix of dataframe columns
:param df: pandas dataframe
:param labels:  String[]
                labels of each dataframe column
:return: matplotlib Axes
        Axes object with the heatmap.
```

5.6.1.4 def plots.plots_raw_data (*df, columns, colors* = "b")

```
Plot raw data of the given dataframe
:param df: DataFrame Wich contains all the data
:param columns: List strings name of the columns to plot
:param colors: color, sequence, or sequence of color, optional, default: 'b'
                c can be a single color format string, or a sequence of color specifications of length N,
                or a sequence of N numbers to be mapped to colors using the cmap and norm specified
                via kwargs (see below).
                Note that c should not be a single numeric RGB or RGBA sequence because that is indistinguisha
                from an array of values to be colormapped. c can be a 2-D array in which the rows are RGB or R
                however, including the case of a single row to specify the same color for all points.
:return:
```

5.6.1.5 def plots.plots_segmentation (*img, labels*)

```
Plot the results of an image segmentation
Example of use:
>>> from skimage.segmentation import slic
>>> img = cv2.imread("path/to/image")
>>> img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
>>> segments = slic(img, n_segments=20, compactness=10, max_iter=100, sigma=2.3)
>>> plots_segmentation(img, segments)
:param img: RGB image
:param labels: 2D or 3D array
                Integer mask indicating segment labels.
:return:
```

Chapter 6

Class Documentation

6.1 LikelihoodGenerator.LikelihoodGenerator Class Reference

Public Member Functions

- def __init__ (self, image)
- def build_your_mask (self, file=None)

Public Attributes

- image
- in_points

6.1.1 Detailed Description

```
Class to help to generate likelihood masks of regions of
interest in images.

Build your likelihood by marking with the mouse the contours of
the roi region.

Keyboard commands:
key a: Submit region selected.
keb b: discard last mouse click

Example of use:
>>> image = "RGB image matrix"
>>> path_output_file = "Path to save the likelihood"
>>> LikelihoodGenerator(image).build_your_mask(path_output_file)
```

6.1.2 Constructor & Destructor Documentation

6.1.2.1 def LikelihoodGenerator.LikelihoodGenerator.__init__ (*self, image*)

```
Init an object to build the interest region of an image
:param image: RGB image
```

6.1.3 Member Function Documentation

6.1.3.1 def LikelihoodGenerator.LikelihoodGenerator.build_your_mask (*self, file =* None)

```
Call this function to init the process of finger_regions selection.
In the process:
Press A (shift + a) to end selection
Press b to cancel the last selection
:return:
```

6.1.4 Member Data Documentation

6.1.4.1 LikelihoodGenerator.LikelihoodGenerator.image

6.1.4.2 LikelihoodGenerator.LikelihoodGenerator.in_points

The documentation for this class was generated from the following file:

- LikelihoodGenerator.py

Chapter 7

File Documentation

7.1 dataframe.py File Reference

Namespaces

- dataframe

Functions

- def dataframe.df_attr_split (df, attr, value, columns)
- def dataframe.df_filter (df, filters)

7.2 functions.py File Reference

Namespaces

- functions

Functions

- def functions.potencia (input, min_n_items=1, max_n_items=None)
- def functions.check_dir (path)

7.3 images.py File Reference

Namespaces

- images

Functions

- def images.binary2RGB (image)
- def images.grays2binary (image_grays)

7.4 LikelihoodGenerator.py File Reference

Classes

- class LikelihoodGenerator.LikelihoodGenerator

Namespaces

- LikelihoodGenerator

7.5 masks.py File Reference

Namespaces

- masks

Functions

- def masks.mask_evaluation (mask, likelihood)
- def masks.mask_sklearn_evaluation (mask, likelihood)
- def masks.masks_coincidence (mask1, mask2, priority="big_mask")
- def masks.mask_onto_mask (mask1, mask2, perc=0.9)
- def masks.mask_2RGB (mask)
- def masks.mask_fill_holes (mask)
- def masks.mask_from_RGB_file (file_mask)
- def masks.mask_bounding_circle (mask)
- def masks.mask_delete_contour_in (mask, region)
- def masks.mask_build_circular (image, circle)
- def masks.mask_build_circular_boolean (image, circle)
- def masks.mask_biggest_connected_component (mask)

7.6 plots.py File Reference

Namespaces

- plots

Functions

- def plots.multiplot (images, filename=None, nrows=2, colorbar=False)
- def plots.plots_correlation_matrix (df, labels=None, absolute=False)
- def plots.plots_segmentation (img, labels)
- def plots.plots_raw_data (df, columns, colors="b")
- def plots.biplot (pca, dat)

7.7 README.md File Reference

Index