Package: SAFARI (via r-universe)

November 17, 2026

Title Shape Analysis for AI-Reconstructed Images

Version 0.1.1

Description Provides functionality for image processing and shape analysis in the context of reconstructed medical images generated by deep learning-based methods or standard image processing algorithms and produced from different medical imaging types, such as X-ray, Computational Tomography (CT), Magnetic Resonance Imaging (MRI), and pathology imaging. Specifically, offers tools to segment regions of interest and to extract quantitative shape descriptors for applications in signal processing, statistical analysis and modeling, and machine learning.

Depends R (>= 3.5.0)

Imports caTools, EBImage, graphics, lattice, png

License GPL (>= 3)

Encoding UTF-8

LazyData true

URL https://github.com/estfernandez/SAFARI

Roxygen list(markdown = TRUE)

RoxygenNote 7.2.1

Config/pak/sysreqs cmake libfftw3-dev make libjpeg-dev libpng-dev libtiff-dev libuv1-dev

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

Date/Publication 2022-07-28 01:08:11 UTC

RemoteUrl https://github.com/estfernan/safari

RemoteRef HEAD

RemoteSha 640b3cb66d34c261e77d0f39ca09ee3688535e71

Contents

SAFARI-package 2
binary.segmentation 2
compute.features 4
rBPS 5
rc.plot 5
read.image 6

Index **8**

`SAFARI-package` *Shape Analysis for AI-Reconstructed Images*

Description

The R package **SAFARI** provides functions for image processing and shape analysis of regions of interest (ROI) in AI-reconstructed images.

Details

The main functions are summarized as follows:

- `read.image`: loads and preprocesses reconstructed binary images for segmentation;
- `binary.segmentation`: ROI segmentation and optional feature extraction;
- `compute.features`: feature extraction for an individual ROI;
- `rc.plot`: visualize reconstructed binary or segmented images.

See the package vignettes for more introduction and demonstration.

`binary.segmentation` *Segmentation Procedure for Binary Images*

Description

Identify and segment the regions of interest (ROI) in a reconstructed binary image, while having the option to extract quantitative shape-based features.

Usage

```
binary.segmentation(
  x,
  id,
  filter = NA,
  k = 3,
  categories = c("none", "geometric", "boundary", "topological")
)
```

Arguments

x	a binary matrix that represents the reconstructed image.
id	a named character vector that contains the ID's pertaining to the sample.
filter	an integer vector that indicates the filtering procedure.
k	an integer that specifies the factor to enlarge the regions.
categories	a character string or vector, see compute.features.

Details

The argument id specifies the ID's corresponding to the sample. A usual example that is used if the argument is an unnamed vector is composed of the following entries:

- cohort: name of cohort the sample belongs to;
- patient.id: unique identifier for patient the sample belongs to;
- slide.id: unique identifier for sample.

The argument filter specifies how the ROI should be filtered. There are two ways to filter them, either by only specifying a minimum net area or by additionally specifying the n largest regions to keep in a two element vector. The default value is NA which sets the minimum net area as one-fourth the largest region.

The argument categories specifies the shape features to compute. The default is "none" which computes no features.

Value

A list object whose components are the following:

- desc: a data.frame of the shape features corresponding to each segmented ROI;
- holes: an integer matrix that contains the holes within each ROI, labeled according to the regions;
- id: a character vector that is identical to the id argument passed;
- k: argument k passed to function;
- n: an integer that indicates the number of segmented regions;
- plg.chains: a list where each component is the polygonal chain of a segmented ROI;
- regions: an integer matrix that contains the segmented ROI, labeled from largest to smallest.

Reconstructed Binary Image

To produce a reconstructed binary image, a greyscale scan or an image easily converted to binary, resulting from different modalities, is converted to a matrix representation, usually by standard image processing algorithms.

The resulting matrix is composed of two integer values that help represent the regions of interest in the scan. Usually, as in the case of pathology imaging, these are empty regions and tumor tissues, where we refer to the integer values as categories. We note that the regions of interest must be represented by 1's and what we consider the empty regions by 0's.

Examples

```
# load libraries
library(SAFARI)

# load sample
data("rBPS")

# segmentation procedure
rBPS <- binary.segmentation(
    rBPS,
    id = c("NLST", "AA00474", "11030"),
    filter = 150,
    categories = c("geometric", "boundary", "topological")
)
```

`compute.features` *Shape Features*

Description

Compute the shape descriptors of the given region, represented by its binary matrix and the polygonal chain of its boundary.

Usage

```
compute.features(
  region,
  plg.chain,
  categories = c("geometric", "boundary", "topological")
)
```

Arguments

`region`	a binary matrix that represents the region.
`plg.chain`	an n-by-2 numeric matrix that represents the boundary.
`categories`	a character string or vector, specifying which features to compute.

Details

The shape descriptors are divided into three categories: geometric, boundary, and topological. The default will compute the features from all categories. We also note that, for the binary matrix, the ones and zeros make up the region of interest and background, respectively.

Value

A numeric vector composed of the shape descriptors.

See Also

See Polygonal Chain for more information on the closed polygonal chain.

---

rBPS *Reconstructed Binary Pathology Slide*

Description

This reconstructed binary image was first represented as a three-class image, prepared using a tumor recognition system (ConvPath) developed by the Quantitative Biomedical Research Center. The original whole-slide image comes from a lung cancer patient in the Lung Screening Study (LSS) subcomponent of NLST. Specifically, this is an image from an H&E-stained slide that was obtained as part of a pathology specimen collection.

Usage

```
data(rBPS)
```

Format

rBPS is a 314-by-224 binary matrix where each entry corresponds to a tissue or region in the H&E image. In our case the ones and zeros indicate an empty region or tumor tissue, respectively.

Source

Original H&E slide available at https://biometry.nci.nih.gov/cdas/nlst/.

References

ConvPath: A software tool for lung adenocarcinoma digital pathological image analysis aided by a convolutional neural network. (2019) EBioMedicine.

---

rc.plot *Reconstructed Image Plot*

Description

Plot a reconstructed image obtained from various types of modalities.

Usage

```
rc.plot(x, type = c("binary", "segments"), publication = FALSE, ...)
```

Arguments

x	an integer matrix that represents the reconstructed image.
type	a character string that specifies the image type.
publication	a logical value, whether to create a figure without key nor scales.
...	additional graphical parameters passed to levelplot.

Details

The argument type specifies the image type with the following options:

- binary: reconstructed binary image, representing a greyscale medical image;
- segments: segmented matrix, where each region is denoted by an integer value.

Value

No return value, called for side effects i.e. plotting.

See Also

binary.segmentation for more information on reconstructed images.

---

read.image *Read a Binary Image*

Description

This function reads a reconstructed binary image, possibly, produced from an X-Ray, CT scan, MRI, etc. and processed by standard image processing algorithms. The image is then pre-processed to facilitate the procedure in the binary.segmentation function.

Usage

```
read.image(file, invert = FALSE, expand.border = 5)
```

Arguments

file	a character string that specifies the image to read.
invert	a logical value that indicates if the B/W colors should be inverted.
expand.border	an integer value that specifies how much to expand the image's border by.

Details

The binary image is pre-processed as follows:

- Check if image contains multiple color channels, non-binary, or empty.
- Invert black and white colors (optional).
- Ensure image is truly binary by converting all values greater than zero to `1`.
- Expand the border around the image with `0`s.
- Rotating the image, if it is not stored as an `RData` file.

Value

An integer matrix that represents the pre-processed image.

See Also

`binary.segmentation` for more information on reconstructed binary images.

Index

∗ **dataset**
rBPS, 5

binary.segmentation, 2, *6, 7*

compute.features, *3*, 4

levelplot, *6*

rBPS, 5
rc.plot, 5
read.image, 6

SAFARI-package, 2