

# Package ‘SuperpixelImageSegmentation’

December 30, 2018

**Type** Package

**Title** Superpixel Image Segmentation

**Version** 1.0.0

**Date** 2018-12-18

**Author** Lampros Mouselimis <mouselimislampros@gmail.com>

**Maintainer** Lampros Mouselimis <mouselimislampros@gmail.com>

**Description** Image Segmentation using Superpixels, Affinity Propagation and Kmeans Clustering. The R code is based primarily on the article “Image Segmentation using SLIC Superpixels and Affinity Propagation Clustering, Bao Zhou, International Journal of Science and Research (IJSR), 2013” <<https://pdfs.semanticscholar.org/6533/654973054b742e725fd433265700c07b48a2.pdf>>.

**License** GPL-3

**Encoding** UTF-8

**LazyData** true

**Depends** R(>= 3.2)

**Imports** Rcpp (>= 0.12.10), R6, OpenImageR

**LinkingTo** Rcpp, RcppArmadillo (>= 0.9.1), ClusterR, OpenImageR

**Suggests** testthat, covr, knitr, rmarkdown

**RoxygenNote** 6.1.0

**NeedsCompilation** yes

**Repository** CRAN

**Date/Publication** 2018-12-30 17:50:03 UTC

## R topics documented:

Image_Segmentation . . . . .	2
<b>Index</b>	<b>5</b>

---

Image_Segmentation	<i>Segmentation of images based on Superpixels, Affinity propagation and Kmeans clustering</i>
--------------------	--

---

### Description

Segmentation of images based on Superpixels, Affinity propagation and Kmeans clustering

### Usage

```
# init <- Image_Segmentation$new()
```

### Arguments

input_image	a 3-dimensional input image (the range of the pixel values should be preferably in the range 0 to 255)
method	a character string specifying the superpixel method. It can be either "slic" or "slico"
superpixel	a numeric value specifying the number of superpixels
kmeans_method	a character string specifying the kmeans method. If not empty ("") then it can be either "kmeans" or "mini_batch_kmeans"
AP_data	a boolean. If TRUE then the affinity propagation image data will be computed and returned
use_median	a boolean. If TRUE then the median will be used rather than the mean value for the inner computations
minib_kmeans_batch	the size of the mini batches
kmeans_num_init	number of times the algorithm will be run with different centroid seeds
kmeans_max_iters	the maximum number of clustering iterations
minib_kmeans_init_fraction	percentage of data to use for the initialization centroids (applies if initializer is <i>kmeans++</i> or <i>optimal_init</i> ). Should be a float number between 0.0 and 1.0.
kmeans_initializer	the method of initialization. One of, <i>optimal_init</i> , <i>quantile_init</i> , <i>kmeans++</i> and <i>random</i> . See details for more information
colour_type	a character string specifying the colour type. It can be one of "RGB", "LAB" or "HSV"
compactness_factor	a numeric value specifying the compactness parameter in case that <i>method</i> is "slic"

<code>adjust_centroids_and_return_masks</code>	a boolean. If TRUE and the <i>kmeans_method</i> parameter is NOT empty ("") then the centroids will be adjusted and image-masks will be returned. This will allow me to plot the masks using the <i>spixel_masks_show</i> method.
<code>sim_normalize</code>	a boolean. If TRUE then the constructed similarity matrix will be normalised to have unit p-norm (see the armadillo documentation for more details)
<code>sim_wL</code>	a numeric value specifying the weight for the "L" channel of the image (see the details section for more information)
<code>sim_wA</code>	a numeric value specifying the weight for the "A" channel of the image (see the details section for more information)
<code>sim_wB</code>	a numeric value specifying the weight for the "B" channel of the image (see the details section for more information)
<code>sim_color_radius</code>	a numeric value specifying the <i>colorradius</i> (see the details section for more information)
<code>delay_display_seconds</code>	a numeric value specifying the seconds to delay the display of the next image (It displays the images consecutively). This parameter applies only if the <i>display_all</i> is set to FALSE ( <i>spixel_masks_show</i> method)
<code>display_all</code>	a boolean. If TRUE then all images will be displayed in a grid ( <i>spixel_masks_show</i> method)
<code>margin_btw_plots</code>	a float number specifying the margins between the plots if the <i>display_all</i> parameter is set to TRUE ( <i>spixel_masks_show</i> method)
<code>verbose</code>	a boolean. If TRUE then information will be printed in the console ( <i>spixel_masks_show</i> method)

## Format

An object of class R6ClassGenerator of length 24.

## Details

*sim\_wL*, *sim\_wA*, *sim\_wB* are the weights of the three channels. They keep balance so as to be consistent with human perception.

The quantity *colorradius* adjusts the number of clusters, and if its value is low, the number of targets would increase, which leads to more detailed segmentation results.

If the *adjust\_centroids\_and\_return\_masks* parameter is set to FALSE then the output *kmeans\_image\_data* will be an RGB image, otherwise it will be a black-and-white image.

*colour\_type* parameter: RGB (Red-Green-Blue), LAB (Lightness, A-colour-dimension, B-colour-dimension) or HSV (Hue, Saturation, Value) colour.

Higher resolution images give better results.

The *affinity propagation* algorithm is used here with default parameter values.

By setting the *sim\_normalize* parameter to TRUE, the affinity propagation algorithm requires less iterations to complete. However, the *colorradius* parameter does not have an effect if the similarity matrix is normalized.

---

 kmeans initializers
 

---

**optimal\_init** : this initializer adds rows of the data incrementally, while checking that they do not already exist in the centroid-matrix

**quantile\_init** : initialization of centroids by using the cumulative distance between observations and by removing potential duplicates

**kmeans++** : kmeans++ initialization. Reference : <http://theory.stanford.edu/~sergei/papers/kMeansPP-soda.pdf> AND <http://stackoverflow.com/questions/5466323/how-exactly-does-k-means-work>

**random** : random selection of data rows as initial centroids

## Methods

```
Image_Segmentation$new()
-----
spixel_segmentation()
-----
spixel_masks_show()
-----
```

## References

<https://pdfs.semanticscholar.org/6533/654973054b742e725fd433265700c07b48a2.pdf>, "Image Segmentation using SLIC Superpixels and Affinity Propagation Clustering", Bao Zhou, 2013, International Journal of Science and Research (IJSR)

## Examples

```
library(SuperpixelImageSegmentation)

path = system.file("images", "BSR_bsds500_image.jpg", package = "SuperpixelImageSegmentation")

im = OpenImageR::readImage(path)

init = Image_Segmentation$new()

num_spix = 10          # for illustration purposes
# num_spix = 600      # recommended number of superpixels

spx = init$spixel_segmentation(input_image = im,
                               superpixel = num_spix,
                               AP_data = TRUE,
                               use_median = TRUE,
                               sim_color_radius = 10)

## Not run:
plt_im = init$spixel_masks_show(display_all = TRUE)

## End(Not run)
```

# Index

\*Topic **datasets**

Image\_Segmentation, [2](#)

Image\_Segmentation, [2](#)