
imgroi Documentation

Release 0.0.2+3.g79c8357.dirty

John Kirkham

Nov 08, 2018

Contents

1	imgroi	3
2	Installation	5
3	Usage	7
4	API	9
5	Contributing	13
6	Indices and tables	17
	Python Module Index	19

Contents:

A Python library for working with ROIs from images.

- Free software: BSD 3-Clause
- Documentation: <https://imgroi.readthedocs.io>.

1.1 Features

- TODO

1.2 Credits

This package was created with [Cookiecutter](#) and the [nanshe-org/nanshe-cookiecutter](#) project template.

2.1 Stable release

To install imgroi, run this command in your terminal:

```
$ pip install imgroi
```

This is the preferred method to install imgroi, as it will always install the most recent stable release.

If you don't have [pip](#) installed, this [Python installation guide](#) can guide you through the process.

2.2 From sources

The sources for imgroi can be downloaded from the [Github repo](#).

You can either clone the public repository:

```
$ git clone git://github.com/jakirkham/imgroi
```

Or download the [tarball](#):

```
$ curl -OL https://github.com/jakirkham/imgroi/tarball/master
```

Once you have a copy of the source, you can install it with:

```
$ python setup.py install
```


CHAPTER 3

Usage

To use imgroi in a project:

```
import imgroi
```


4.1 imgroi package

4.1.1 Submodules

imgroi.core module

`imgroi.core.find_contours` (*img*)

Takes an image and extracts contours from the mask.

Parameters `a_image` (*numpy.ndarray*) – takes an image.

Returns an array with contours.

Return type (*numpy.ndarray*)

Examples

```
>>> numpy.set_printoptions(legacy="1.13")
```

```
>>> a = numpy.array([[ True,  True, False],
...                  [False, False, False],
...                  [ True,  True,  True]], dtype=bool)
```

```
>>> find_contours(a)
array([[ True,  True, False],
       [False, False, False],
       [ True,  True,  True]], dtype=bool)
```

```
>>> find_contours(numpy.eye(3))
array([[ 1.,  0.,  0.],
```

(continues on next page)

(continued from previous page)

```
[ 0.,  1.,  0.],
 [ 0.,  0.,  1.]])
```

```
>>> a = numpy.array([
...     [False, False,  True, False, False, False,  True],
...     [ True, False, False, False,  True, False, False],
...     [ True,  True, False,  True,  True, False,  True],
...     [ True, False, False,  True,  True, False, False],
...     [ True, False, False, False, False, False, False],
...     [False,  True, False, False, False, False,  True],
...     [False,  True,  True, False, False, False, False]
... ], dtype=bool)
```

```
>>> find_contours(a)
array([[False, False,  True, False, False, False,  True],
       [ True, False, False, False,  True, False, False],
       [ True,  True, False,  True,  True, False,  True],
       [ True, False, False,  True,  True, False, False],
       [ True, False, False, False, False, False, False],
       [False,  True, False, False, False, False,  True],
       [False,  True,  True, False, False, False, False]], dtype=bool)
```

`imgroi.core.label_mask_stack(new_masks, dtype=None)`

Takes a mask stack and replaces them by the max of an enumerated stack. In other words, each mask is replaced by a consecutive integer (starts with 1 and proceeds to the length of the given axis (0 by default)). Afterwards, the max is taken along the given axis.

Parameters

- **new_masks** (*numpy.ndarray*) – masks to enumerate
- **dtype** (*type*) – type to use for the label matrix (default is int).

Returns an enumerated stack.

Return type (*numpy.ndarray*)

Examples

```
>>> numpy.set_printoptions(legacy="1.13")
```

```
>>> label_mask_stack(
...     numpy.array([[1, 0, 0, 0],
...                  [0, 0, 0, 0],
...                  [0, 0, 0, 0],
...                  [0, 0, 0, 0]],
...
...                 [[0, 0, 0, 0],
...                  [0, 1, 0, 0],
...                  [0, 0, 0, 0],
...                  [0, 0, 0, 0]],
...
...                 [[0, 0, 0, 0],
...                  [0, 0, 0, 0],
...                  [0, 0, 1, 0]]))
```

(continues on next page)

(continued from previous page)

```
...         [0, 0, 0, 0]],
...
...         [[0, 0, 0, 0],
...         [0, 0, 0, 0],
...         [0, 0, 0, 0],
...         [0, 0, 0, 1]]], dtype=bool)
... )
array([[1, 0, 0, 0],
       [0, 2, 0, 0],
       [0, 0, 3, 0],
       [0, 0, 0, 4]])
```


Contributions are welcome, and they are greatly appreciated! Every little bit helps, and credit will always be given. You can contribute in many ways:

5.1 Types of Contributions

5.1.1 Report Bugs

Report bugs at <https://github.com/jakirkham/imgroi/issues>.

If you are reporting a bug, please include:

- Your operating system name and version.
- Any details about your local setup that might be helpful in troubleshooting.
- Detailed steps to reproduce the bug.

5.1.2 Fix Bugs

Look through the GitHub issues for bugs. Anything tagged with “bug” and “help wanted” is open to whoever wants to implement it.

5.1.3 Implement Features

Look through the GitHub issues for features. Anything tagged with “enhancement” and “help wanted” is open to whoever wants to implement it.

5.1.4 Write Documentation

imgroi could always use more documentation, whether as part of the official imgroi docs, in docstrings, or even on the web in blog posts, articles, and such.

5.1.5 Submit Feedback

The best way to send feedback is to file an issue at <https://github.com/jakirkham/imgroi/issues>.

If you are proposing a feature:

- Explain in detail how it would work.
- Keep the scope as narrow as possible, to make it easier to implement.
- Remember that this is a volunteer-driven project, and that contributions are welcome :)

5.2 Get Started!

Ready to contribute? Here's how to set up *imgroi* for local development.

1. Fork the *imgroi* repo on GitHub.
2. Clone your fork locally:

```
$ git clone git@github.com:your_name_here/imgroi.git
```

3. Install your local copy into an environment. Assuming you have conda installed, this is how you set up your fork for local development (on Windows drop *source*). Replace “<some version>” with the Python version used for testing.:

```
$ conda create -n imgroienv python="<some version>"
$ source activate imgroienv
$ python setup.py develop
```

4. Create a branch for local development:

```
$ git checkout -b name-of-your-bugfix-or-feature
```

Now you can make your changes locally.

5. When you're done making changes, check that your changes pass flake8 and the tests, including testing other Python versions:

```
$ flake8 imgroi tests
$ python setup.py test or py.test
```

To get flake8, just conda install it into your environment.

6. Commit your changes and push your branch to GitHub:

```
$ git add .
$ git commit -m "Your detailed description of your changes."
$ git push origin name-of-your-bugfix-or-feature
```

7. Submit a pull request through the GitHub website.

5.3 Pull Request Guidelines

Before you submit a pull request, check that it meets these guidelines:

1. The pull request should include tests.
2. If the pull request adds functionality, the docs should be updated. Put your new functionality into a function with a docstring, and add the feature to the list in README.rst.
3. The pull request should work for Python 2.7, 3.4, 3.5, and 3.6. Check https://travis-ci.org/jakirkham/imgroi/pull_requests and make sure that the tests pass for all supported Python versions.

5.4 Tips

To run a subset of tests:

```
$ python -m unittest tests.test_imgroi
```


CHAPTER 6

Indices and tables

- `genindex`
- `modindex`
- `search`

i

- `imgroi`, 9
- `imgroi.core`, 9

F

`find_contours()` (in module `imgroi.core`), [9](#)

I

`imgroi` (module), [9](#)

`imgroi.core` (module), [9](#)

L

`label_mask_stack()` (in module `imgroi.core`), [10](#)