
facehugger Documentation

Release 0.1.5

Diego Navarro Mellen

November 05, 2013

Contents

1	facehugger	3
1.1	Requirements	3
1.2	Features	3
1.3	Quickstart	3
1.4	Why this name?	3
1.5	Did you like it?	3
2	Installation	5
3	Usage	7
3.1	Rescaling resulting face crops	7
4	Contributing	9
4.1	Types of Contributions	9
4.2	Get Started!	10
4.3	Pull Request Guidelines	10
4.4	Tips	11
5	Credits	13
5.1	Development Lead	13
5.2	Contributors	13
6	History	15
6.1	0.1.0 (2013-10-30)	15
6.2	0.1.3 (2013-10-30)	15
6.3	0.1.4 (2013-10-30)	15
7	Indices and tables	17

Contents:

facehugger

Extracts faces from an image

- Free software: BSD license
- Documentation: <http://facehugger.rtfid.org>.

1.1 Requirements

- SimpleCV

System requirements for Ubuntu

```
sudo apt-get install ipython python-opencv python-scipy python-numpy python-pygame python-setuptools
```

1.2 Features

- Extracts faces and saves the cropped images to a folder
- You can use it in your app as a library

1.3 Quickstart

```
$ sudo pip install facehugger
$ facehugger -i /path/to/image_with_faces.jpg -o /path/to/dir_to_save_faces
```

1.4 Why this name?

1.5 Did you like it?

Installation

At the command line:

```
$ easy_install facehugger
```

Or, if you have virtualenvwrapper installed:

```
$ mkvirtualenv facehugger  
$ pip install facehugger
```

Usage

To use facehugger in a project:

```
from facehugger import get_faces

faces = get_faces('/some/path/to_image.jpg')
```

To use facehugger from command line:

```
$ sudo pip install facehugger
$ facehugger -i /path/to/image_with_faces.jpg -o /path/to/dir_to_save_faces
```

3.1 Rescaling resulting face crops

New in version 0.1.5: Added *–rescale-face-crop* parameter that receives an integer representing percentage to scale up the crop

```
$ facehugger -i /path/to/image_with_faces.jpg -o /path/to/dir_to_save_faces --rescale-face-crop 20
```

Contributing

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

4.1 Types of Contributions

4.1.1 Report Bugs

Report bugs at <https://github.com/dnmellen/facehugger/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.

4.1.2 Fix Bugs

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

4.1.3 Implement Features

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

4.1.4 Write Documentation

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

4.1.5 Submit Feedback

The best way to send feedback is to file an issue at <https://github.com/dnmellen/facehugger/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 :)

4.2 Get Started!

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

1. Fork the *facehugger* repo on GitHub.

2. Clone your fork locally:

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

3. Install your local copy into a virtualenv. Assuming you have *virtualenvwrapper* installed, this is how you set up your fork for local development:

```
$ mkvirtualenv facehugger
$ cd facehugger/
$ 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 with *tox*:

```
$ flake8 facehugger tests
    $ python setup.py test
$ tox
```

To get *flake8* and *tox*, just *pip* install them into your virtualenv.

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.

4.3 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. Check https://travis-ci.org/dnmellen/facehugger/pull_requests and make sure that the tests pass for all supported Python versions.

4.4 Tips

To run a subset of tests:

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

Credits

5.1 Development Lead

- Diego Navarro Mellen <dnmellen@gmail.com>

5.2 Contributors

None yet. Why not be the first?

History

6.1 0.1.0 (2013-10-30)

- First release on PyPI.

6.2 0.1.3 (2013-10-30)

- First working release.

6.3 0.1.4 (2013-10-30)

- Updated docs.

Indices and tables

- *genindex*
- *modindex*
- *search*