

---

# **Acoustic Scene Classification Documentation**

***Release 0.3.1***

**Matthieu Berjon**

**Nov 24, 2019**



---

## Contents

---

|          |   |           |
|----------|---|-----------|
| <b>1</b> | <b>Acoustic Scene Classification</b>                                | <b>3</b>  |
| 1.1      | Dataset . . . . .   | 3         |
| 1.2      | Features . . . . .  | 3         |
| 1.3      | Credits . . . . .   | 3         |
| <b>2</b> | <b>Installation</b>   | <b>5</b>  |
| 2.1      | Stable release . . . . .  | 5         |
| 2.2      | From sources . . . . .  | 5         |
| <b>3</b> | <b>Usage</b>  | <b>7</b>  |
| 3.1      | Command line . . . . .  | 7         |
| 3.2      | Library import . . . . .  | 7         |
| <b>4</b> | <b>Configuration</b>  | <b>9</b>  |
| <b>5</b> | <b>asc</b>  | <b>11</b> |
| 5.1      | asc package . . . . .   | 11        |
| <b>6</b> | <b>Contributing</b>   | <b>13</b> |
| 6.1      | Types of Contributions . . . . .                                    | 13        |
| 6.2      | Get Started! . . . . .  | 14        |
| 6.3      | Pull Request Guidelines . . . . .                                   | 15        |
| 6.4      | Tips . . . . .  | 15        |
| <b>7</b> | <b>Credits</b>  | <b>17</b> |
| 7.1      | Development Lead . . . . .  | 17        |
| 7.2      | Contributors . . . . .  | 17        |
| <b>8</b> | <b>History</b>  | <b>19</b> |
| 8.1      | 0.3.1 (2017-08-22) . . . . .  | 19        |
| 8.2      | 0.2.3 (2017-08-07) . . . . .  | 20        |
| 8.3      | 0.2.2 (2017-07-31) . . . . .  | 21        |
| 8.4      | 0.2.1 (2017-07-31) . . . . .  | 21        |
| 8.5      | 0.2.0 (2017-07-31) . . . . .  | 22        |
| 8.6      | 0.1.0 (2017-07-25) . . . . .  | 23        |
| <b>9</b> | <b>FAQ</b>  | <b>25</b> |
| 9.1      | Error at the installation on Fedora/Debian and python 2.7 . . . . . | 25        |

|     |   |           |
|-----|---|-----------|
| 9.2 | tkinter.TclError: no display name and no \$DISPLAY environment variable . . . . . | 25        |
|     | <b>Python Module Index</b>  | <b>27</b> |
|     | <b>Index</b>  | <b>29</b> |

Contents:



---

## Acoustic Scene Classification

---

|                     |  |
|---------------------|--|
| <b>Development</b>  |  |
| <b>Last release</b> |  |
| <b>PyPI status</b>  |  |

Acoustic Scene Auditory (ASC) using Convolutional Neural Network (CNN) is a project being part of the Machine Learning Nanodegree program given by Udacity. For a description of the proposal, you can refer to its [web version](#).

### 1.1 Dataset

The dataset can be downloaded on the [Zenodo](#) server.

### 1.2 Features

- TODO

### 1.3 Credits

Project created by [Matthieu Berjon](#) and based on the work of Simone Battaglini, Ludovick Lepauloux and Nicholas Evans.

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





### 2.1 Stable release

To install Acoustic Scene Classification, run this command in your terminal:

```
$ pip install asc
```

This is the preferred method to install Acoustic Scene Classification, 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.

#### 2.1.1 Requirements

In order to properly work you need to satisfy several dependencies:

- tkinter (python 2 or 3 depending of your system)
- libsndfile
- python-dev

For example on Debian:

```
$ sudo apt-get install libsndfile python-tk python-dev
```

If you still have issues at the installation, you can still refer to the [FAQ](#) or post an issue on the [Github repo](#).

### 2.2 From sources

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

You can either clone the public repository:

```
$ git clone git://github.com/mattberjon/asc
```

Or download the [tarball](#):

```
$ curl -OL https://github.com/mattberjon/asc/tarball/master
```

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

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

It is possible to use ASC loading the library and integrates it into your current project. You can as well using the command line to use it as it is.

### 3.1 Command line

Several commands are available and can be obtained using:

```
$ asc --help
```

In order to setup properly the project, I would recommend to run the setup command as follow:

```
$ asc setup
```

You'll be asked to enter the root path where you want to store the data. Then you can download the data using the command:

```
$ asc getdata
```

You can take a coffee and wait until archives are downloaded and the audio files extracted into the audio folder.

### 3.2 Library import

To use Acoustic Scene Classification in a project:

```
import asc
```



## CHAPTER 4

---

### Configuration

---

| Parameters       | Values                 | Description  |
|------------------|------------------------|--|
| audio.samplerate | (int) default=44100    | Samplerate of the project in Hertz                           |
| audio.mel_bands  | (int) default=128      | Number of mel bands for the filter                           |
| audio.fmax       | (int) default=22050    | Frequency max in Hertz for the mel filter decomposition      |
| path.tmp         | (str) default='/tmp'   | Path to the temporary download folder                        |
| path.data        | (str) default='~/'     | Path to the data folder                                      |
| path.url_list    | (str) no default value | Path to the file containing the URLs of the data to download |



## 5.1 asc package

### 5.1.1 Submodules

### 5.1.2 asc.audio module

### 5.1.3 asc.cli module

### 5.1.4 asc.data module

Data module

This module lets the program to download the data from the server.

**class** `asc.data.Data`

Bases: `object`

Data collection.

**clear\_zip** (*url\_list*, *tmp\_dir*)

Clear the archives

Delete the downloaded archives.

**Args:** *url\_list* (str): list of strings containing the urls *tmp\_dir* (str): path where are store the archives

**download** (*url\_list*, *dest\_dir*)

Download data from a list of URLs

Download data from a list of URLs and display a progress according to the size of the file.

**Args:** *url\_list*(list): list containing URLs of files to download. *dest\_dir* (str): path to where download the data.

### **file\_to\_list** (*filename*)

Parse a file and transform it into a list

Parse a file line by line to make a list with the urls contained inside.

**Args:** filename (str): path to the file containing the URLs.

### **move\_files** (*url\_list*, *origin\_dir*, *dest\_dir*)

Move the audio files into the data folder

Move the audio folder created by the unzipped archives into the data root folder. The function can specify the origin and destination folder but assumes that the audio files are located into an folder called 'audio' and can't be changed.

**Args:** url\_list (list): list of all the archive URLs. origin\_dir (str): directory of origin. dest\_dir (str): directory of destination.

**Returns:** None

**Todo:** The fact that the 'audio' folder is hard coded makes this function rather bad and should be corrected in the future.

### **unzip\_data** (*url\_list*, *origin\_dir*, *dest\_dir*)

Unzip data files

Unzip data files given a list of file names, the path where they are store and where they will be unzipped.

**Args:** url\_list (str): list of strings containing the urls origin\_dir (str): directory where are stored the files dest\_dir (str): directory where the files will be extracted

## 5.1.5 asc.utils module

## 5.1.6 Module contents

Top-level package for Acoustic Scene Classification.



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

## 6.1 Types of Contributions

### 6.1.1 Report Bugs

Report bugs at <https://github.com/mattberjon/asc-cnn/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.

### 6.1.2 Fix Bugs

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

### 6.1.3 Implement Features

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

## 6.1.4 Write Documentation

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

## 6.1.5 Submit Feedback

The best way to send feedback is to file an issue at <https://github.com/mattberjon/asc-cnn/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 :)

## 6.2 Get Started!

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

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

```
$ git clone git@github.com:your_name_here/asc-cnn.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 asc
$ cd asc/
$ 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 asc tests
$ python setup.py test or 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.

## 6.3 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 2.6, 2.7, 3.3, 3.4 and 3.5, and for PyPy. Check [https://travis-ci.org/mattberjon/asc-cnn/pull\\_requests](https://travis-ci.org/mattberjon/asc-cnn/pull_requests) and make sure that the tests pass for all supported Python versions.

## 6.4 Tips

To run a subset of tests:

```
$ py.test tests.test_asc
```



### 7.1 Development Lead

- Matthieu Berjon <[matthieu@berjon.net](mailto:matthieu@berjon.net)>

### 7.2 Contributors

None yet. Why not be the first?



## 8.1 0.3.1 (2017-08-22)

### 8.1.1 Added

- [Issue #8](#): config file feature
- [Issue #11](#): Samplerate setting up through CLI
- [Issue #10](#): Spectrogram calculation
- [Issue #24](#): Add a 'config' subcommand to the CLI
- [Issue #25](#): Read a value from the config file
- [Issue #21](#): Script to verify the samplerate sanity of the database
- [Issue #30](#): Function to transform milliseconds to samples

### 8.1.2 Changed

- [Issue #23](#): Update of the documentation for the installation process
- [Issue #26](#): Update of the CLI subcommand 'getdata'
- [Issue #32](#): Exception handling in config reading

### 8.1.3 Deprecated

Nothing.

### 8.1.4 Removed

Nothing.

### 8.1.5 Fixed

- [Issue #9](#): trailing slash on data path
- [Issue #13](#): Update of the config file parameters
- [Issue #15](#): Crash when setting up the samplerate
- [Issue #22](#): Update of the dependency list
- [Issue #25](#): Update of the config file
- Update of docstrings
- Update of the documentation

### 8.1.6 Security

- Update of the Coveralls library from 1.1 to 1.2.0

## 8.2 0.2.3 (2017-08-07)

### 8.2.1 Added

- Travis config file
- pytest suite
- CLI tests
- [issue #5](#): Package coverage for the development setup
- [issue #6](#): adding of a `clear_zip()` to clean the archive files

### 8.2.2 Changed

- Python 3.3 testing removed

### 8.2.3 Deprecated

- Nothing

### 8.2.4 Removed

- Nothing

### 8.2.5 Fixed

- [Issue #4](#): invalid functools dependency
- [Issue #7](#): update of docstring `unzip_data()`



### 8.2.6 Security

- Nothing

## 8.3 0.2.2 (2017-07-31)

### 8.3.1 Added

- nothing

### 8.3.2 Changed

- Updated of the ChangeLog (HISTORY.rst)

### 8.3.3 Deprecated

- nothing

### 8.3.4 Removed

- nothing

### 8.3.5 Fixed

- nothing

### 8.3.6 Security

- nothing

## 8.4 0.2.1 (2017-07-31)

### 8.4.1 Added

- nothing

### 8.4.2 Changed

- nothing

### 8.4.3 Deprecated

- nothing

#### 8.4.4 Removed

- nothing

#### 8.4.5 Fixed

- unzip\_data() url list issue
- download of temporary files in the right directory

#### 8.4.6 Security

- nothing

### 8.5 0.2.0 (2017-07-31)

#### 8.5.1 Added

- Adding of a documentation (with docstrings)
- CLI command to download and unzip data automatically
- creation of a python package
- configuration of Tox
- download() method in data class

#### 8.5.2 Changed

- Use of RST instead of markdown for all the documentation
- development packages are now in requirements\_dev.txt

#### 8.5.3 Deprecated

- nothing

#### 8.5.4 Removed

- nothing

#### 8.5.5 Fixed

- source files satisfy PEP8
- bug fix on getdata cli

### 8.5.6 Security

- Update of all packages to their latest versions

## 8.6 0.1.0 (2017-07-25)

- First release as a package.



## 9.1 Error at the installation on Fedora/Debian and python 2.7

The package is using several libraries that need to be compiled in order to work properly as they aren't 100% python. If you get an error such as

```
gcc: error: /usr/lib/rpm/redhat/redhat-hardened-cc1: No such file or directory
```

You can try to resolve the issue by installation the package *redhat-rpm-config*, or *build-essential* on Debian based flavours.

## 9.2 tkinter.TclError: no display name and no \$DISPLAY environnement variable

If for some reasons you run this program on machine without XWindow, you will most probably encounter this error. It exists a workaround by specifying to Matplotlib which backend you want to use. Most probably you'll use the default backend called Agg. In order to do that, you need to edit the *matplotlibrc* file located on GNU/Linux in *~/.config/matplotlib/matplotlibrc*.

You will need to edit or add the following line:

```
backend      : Agg
```

If you don't use GNU/Linux or want further information about this particular setup, I advise you to visit the [Matplotlib documentation](#).



### **a**

`asc`, [12](#)

`asc.data`, [11](#)





## A

`asc` (*module*), [12](#)

`asc.data` (*module*), [11](#)

## C

`clear_zip()` (*asc.data.Data method*), [11](#)

## D

`Data` (*class in asc.data*), [11](#)

`download()` (*asc.data.Data method*), [11](#)

## F

`file_to_list()` (*asc.data.Data method*), [11](#)

## M

`move_files()` (*asc.data.Data method*), [12](#)

## U

`unzip_data()` (*asc.data.Data method*), [12](#)