1. If your PR introduces a change in functionality, make sure you start by opening an issue to discuss whether the

change should be made, and how to handle it. This is of course, not needed for simple bug fixes.

2. Make sure any new function or class you introduce has proper docstrings. Make sure any code you touch still has

up-to-date docstrings and documentation. \*\*Docstring style should be respected.\*\*

In particular, they should be formatted in MarkDown, and there should be sections for `Arguments`, `Returns`, `Raises` (if applicable).

Look at other docstrings in the codebase for examples.

3. Write tests. Your code should have full unit test coverage and should run with 'theano' and 'tensorflow' backends

with 'channels\_first' and 'channels\_last' image\_data\_format(s). To run locally:

- You will need to install the test requirements: `pip install -e .[tests]`.

- Install PEP8 packages: `pip install pep8 pytest-pep8 autopep8`

- Run `py.test --pep8 -m pep8 -n0` to verify PEP8 syntax check.

- Run tests using `cd tests/`, `py.test`

- You can automatically fix some PEP8 error by running: `autopep8 -i --select <errors> <FILENAME>`.

For example: `autopep8 -i --select E128 tests/vis/test\_attention.py`

4. When committing, use appropriate, descriptive commit messages.

5. Update the documentation. If introducing new functionality, make sure you include code snippets demonstrating the usage of your new feature.