# **How to contribute**

If you have any questions that are beyond the scope of this help file, please feel free to ask for help in the [forum](https://forum.medusabci.com/) or [contact with us](mailto:support@medusabci.com).

MEDUSA has been developed by researchers and engineers from the Brain-Computer Interface Lab within the [Biomedical Engineering Group](http://www.gib.tel.uva.es/) of the University of Valladolid. The team open sourced MEDUSA in the hope of sharing our tools and knowledge with the external community to encourage collaboration between researchers and industry. MEDUSA is growing into a thriving ecosystem of projects, but our goal is still to make neurotechnology accessible to anyone, anywhere. That said, the MEDUSA ecosystem can only become what we envision with the contributions of a wider neurotech community. If you share our vision, we encourage you to contribute in any way you can.  Thank you so much for your enthusiasm and your work—we appreciate everything you do!

There are different ways to contribute to MEDUSA© Kernel. This guide includes everything you need to get started. The main ways to contribute are:

* Contribute code
* Improvements to API documentation
* Create or improve tutorials
* Report issues and make suggestions in the official repository

# Contribute code

Whether you are adding new signal processing method or proposing a major design change in the API, this portion of the contributor guide will help you get started. Thank you for work and interest in improving MEDUSA© Kernel.

First, you need to know that new features, bug fixes, and any other changes to the code base are subject to code review. Reviewing code contributed to the project as pull requests is a crucial component of MEDUSA development. We encourage anyone to start reviewing code submitted by other developers, especially if the feature is something that you are likely to use.

Here are some questions to keep in mind during the code review process:

* Do we want this in MEDUSA? Is it likely to be used? Do you, as a MEDUSA user, like the change and intend to use it? Is this change in the scope of MEDUSA? Will the cost of maintaining a new feature be worth its benefits?
* Is the code consistent with the MEDUSA API? Are public functions, classes, and parameters well-named and intuitively designed?
* Does it include documentation? Are all public functions, classes, parameters, return types, and stored attributes named according to MEDUSA conventions and clearly documented? Is the new functionality described in MEDUSA documentation and illustrated with examples, whenever possible? Does the documentation render properly?
* Is the code human-readable? Is it low on redundancy? Should variable names be improved for clarity or consistency? Should comments be added? Should any comments be removed as unhelpful or extraneous?
* Is the code efficient? Could it be rewritten easily to run more efficiently?
* Is the code backwards compatible with previous versions of MEDUSA?
* Will the new code add new dependencies on other libraries? External dependencies should be reduced to the minimum.

## **Contribution workflow**

Code contributions—bug fixes, new development, test improvement—all follow a GitHub-centered workflow. To participate in TensorFlow development, set up a GitHub account. Then:

1. Fork the [official repository](https://github.com/medusabci/medusa-kernel) using the Fork button in GitHub. This will create a copy of the repo, under your username. For more details on how to fork a repository see this [guide](https://help.github.com/articles/fork-a-repo/).
2. Clone down the repo to your local system.
3. Create a new branch to hold your work.

$ git checkout -b new-branch-name

1. Work on your new code and documentation.
2. Commit your changes.

$ git add -A

$ git commit -m "commit message here"

1. Push your changes to your GitHub repo.

$ git push origin branch-name

1. Open a Pull Request (PR). Go to the original project repo on GitHub. There will be a message about your recently pushed branch, asking if you would like to open a pull request. Follow the prompts, compare across repositories, and submit the PR. This will send an email to the committers. You may want to consider sending an email to the mailing list for more visibility. For more details, see the [GitHub guide on PRs](https://help.github.com/articles/creating-a-pull-request-from-a-fork).
2. Maintainers and other contributors will review your PR. Please participate in the conversation and try to make any requested changes. Once the PR is approved, the code will be merged.

# Contribute to API documentation

MEDUSA welcomes documentation contributions—if you improve the documentation, you improve MEDUSA itself. The documentation in this web page falls into the following categories:

* API reference —The [API reference docs](https://www.tensorflow.org/api_docs/) are generated from docstrings in the [MEDUSA source code](https://github.com/tensorflow/tensorflow).
* Narrative documentation —These are [tutorials](https://www.tensorflow.org/tutorials), [guides](https://www.tensorflow.org/guide), and other writing that's not part of the MEDUSA code. This documentation is in the [medusa-tutorials](https://github.com/medusabci/medusa-tutorials) GitHub repository.

## **Contributions to the API reference**

Contributions to the API reference are made by changing the docstrings in the MEDUSA source code. Follow the steps in the contribute code tutorial to update the documentation. MEDUSA docstrings are built using Sphinx using the numpydocs syntax. Any contribution must follow this syntax. To know more about this numpydocs, see the [official website.](https://numpydoc.readthedocs.io/en/latest/format.html)

## **Contributions to the narrative documentation**

To contribute to the narrative documentation, which includes Jupyter notebooks with examples that show different parts of the MEDUSA© Kernel library, you need to update the [medusa-tutorials](https://github.com/medusabci/medusa-tutorials) repository. Any new tutorial added to this repository will be displayed automatically in this documentation. To contribute to the repository, the process is the same that for the source code. In this case, you will need to fork the [medusa-tutorials repository](https://github.com/medusabci/medusa-tutorials).

# Contribute by reporting issues suggestions

If you don’t have the time or knowledge to contribute source code to the project, you can also report issues and make improvement suggestions. This task is equally appreciated and will speed up the improvement of MEDUSA© Kernel. To report issues and suggestions, use the issues section of the [official repository](https://github.com/medusabci/medusa-kernel/issues).