

The OpenCV repository has several branches with different contribution policies.

Common rules for all branches

- Your development branch name must differ from the names of branches in the upstream OpenCV repository, i.e. your branch must **NOT** be named *2.4*, *3.4*, *4.x*, *5.x*, or *master/next* (it's a technical requirement specific to our continuous integration system to properly test multi-repository patches with `opencv_extra/opencv_contrib`)
- Multiple related commits should be squashed into one. A pull request must contain either a single commit, or several unrelated commits (e.g, commit with test + commit with code fix)

Branches and contribution policies in OpenCV

Please target Pull Requests (PR) to the right branches. Quick summary of the rules is presented in the table below. More formal description is in the sections after the table.

Which branch should I target my PR?

Policy	5.x	4.x	3.4
Preserve API compatibility	:heavy_check_mark:	:heavy_check_mark:	:heavy_check_mark:
Preserve ABI compatibility	:x:	:x::grey_exclamation:	:heavy_check_mark:
Change applicable only to this branch	:heavy_check_mark:	:heavy_check_mark:	:heavy_check_mark:
Bugfix / minor fix	:point_right:	:point_right:	:heavy_check_mark:
Optimization	:point_right:	:point_right:	:heavy_check_mark:
Small feature	:point_right:	:heavy_check_mark:	:x:
Large feature	:heavy_check_mark:	:x:	:x:
Branch alias (do not use!)	next	master	-

- :grey_exclamation: - avoid major breakages
- :point_right: - rebase to previous branch if patch is applicable, it will be ported to other branches by the Core team in a week or two

5.x

This is the development branch for upcoming 5.x releases.

- API compatibility must be preserved

- If your pull request contains a bug fix which is also applicable to the *3.4* or *4.x* branch, you should choose that branch as “base”
- If you’ve already created pull request based on the *5.x* branch, but it is also applicable to *4.x*, you will be asked to rebase it to *4.x/3.4* branches, see the instruction in the following section.

4.x

This is the branch for 4.x releases.

- API compatibility must be preserved
- Incompatible improvements or large features should be targeted to *5.x* branch
- Compatible improvements or small features should go to *4.x* branch
- If your pull request contains a bug fix which is also applicable to the *3.4* branch, you should choose that branch as “base”
- If you’ve already created pull request based on the *4.x* branch, but it is also applicable to *3.4*, you will be asked to rebase it to *3.4*, see the instruction in the following section.
- We will merge changes from *4.x* into *5.x* regularly (weekly/bi-weekly) There is no regular process for backporting patches from *5.x* branch

3.4

This is the branch for 3.4.x releases. This release series is in the maintenance status. Releases with binary artifacts are not planned.

- ABI compatibility must be preserved
- Target this branch in case of these patches:
 - bug fixes
 - optimizations
 - documentation improvements
 - and other patches which are applicable for the *3.4* branch
- Improvements or features should be targeted to *4.x* or *5.x* branches
- We will merge changes from *3.4* into *4.x* regularly (weekly/bi-weekly) So if your pull request contains a bug fix and it is applicable to several branches, you should choose *3.4* branch as base There is no regular process for backporting patches from *4.x*, *5.x* branches

EOL branches

EOL branches: - “2.4”: this is the branch for 2.4.x releases. This release series is EOL in 2020. CI is turned off, so no more patches are accepted to 2.4 branch.

master / next

“master” branch is a shadow copy of 4.x development branch kept for compatibility purposes. After 5.0 release this branch will shadow *5.x* branch. “next” branch

is a shadow copy of 5.x development branch kept for compatibility purposes. It will be removed after 5.0 release. Merge to this branches is prohibited. Use 4.x / 5.x as target branches for your PRs.

Rebasing a pull request between branches

If you can not do this by yourself, please ask maintainers for help.

Example of rebasing from 4.x branch to 3.4 branch: * do not close the existing pull request * change the “base” branch of the pull request: * open PR on GitHub in your browser * press the “Edit” button near the pull request title * choose “3.4” from the dropdown list * rebase your commits from 4.x onto 3.4: * `git checkout <your-branch>` * (optional) create a backup branch: `git branch <your-branch>-backup` * `git remote update upstream` (assuming *upstream* is pointing to the `opencv/opencv` GitHub repository) * `git rebase -i --onto upstream/3.4 upstream/4.x` * an editor will be opened. Please check the list of commits - there should be only your commits - save and exit * `git push --force origin <your-branch>` (assuming *origin* is pointing to `<your-username>/opencv` forked GitHub repository)

Related articles

- [GitHub Flow guide](#)
- [Forking Projects guide](#)
- [ABI Compliance Checker tool](#)