

QA in OpenCV

C++ tests

You can start from the section 9 in OpenCV Coding Style Guide.

Same tests are used on every supported platform: Windows, Linux, MacOS and Android. Majority of testing is performed using C++ API, because it is the preferred language in the OpenCV project. These tests are implemented on top of Google Test unit-testing framework.

There are two major types of C++ tests: *accuracy/regression* tests and *performance* tests. Each module can have two test binaries: `opencv_test_<MODULE_NAME>` and `opencv_perf_<MODULE_NAME>`. These applications can be built for every supported platform (Win/Lin/Mac/Android).

Please note that you should set `OPENCV_TEST_DATA_PATH` environment variable (see section 9 in OpenCV Coding Style Guide). On Android you should copy test data to the `/sdcard/opencv_testdata` directory:

```
git clone git://github.com/opencv/opencv_extra.git .
adb push ./testdata /sdcard/opencv_testdata
```

Accuracy Tests

OpenCV library is actively developed, hence a set of regression tests is required. At the same time conformance between different platforms (both software and hardware) should be controlled, this is vital for keeping computer vision algorithms portable. For these two purposes OpenCV contains so-called accuracy tests. These unit-tests execute OpenCV functions with different parameters (input/output matrices sizes, number of channels, algorithm parameters and finally input data) and test it against some reference/desired output.

Every OpenCV module (*core*, *imgproc*, etc) has a set of accuracy tests, and their sources can be found in OpenCV repository in `modules/<MODULE_NAME>/test` directories. For example `modules/core/test` and `modules/imgproc/test`.

Performance Tests

Performance tests sources are located in the `modules/<MODULE_NAME>/perf` directories. For example `modules/core/perf` and `modules/imgproc/perf`.

More technical information about performance tests can be found in the following documents: - [\[\[How to read/write perf tests|HowToWritePerfTests\]\]](#) - [\[\[How to execute/analyse perf tests|HowToUsePerfTests\]\]](#)

Java Tests

- [\[\[Java testing on Android|Android_Java_API_tests\]\]](#)
- Java testing on desktop:

- Set environment variable `OPENCV_TEST_DATA_PATH`
- Run `python <opencv>/modules/ts/misc/run.py . -a -t java` from the build directory

BuildBot

OpenCV's continuous integration system is available here: <http://pullrequest.opencv.org>. There are also separate build-slaves for documentation, distribution package, test coverage calculation and some others. Additionally a small subset of builders is run on every commit to the OpenCV source code repository.

Documentation

Documentation update check list for public releases:

- Check all images manually. Update images, that include OpenCV version, if it is needed by context;
- Check all .rst files. Find docs with links to old versions of OpenCV: 2.4.3, 2.4.2, 2.4.1, 2.4.0, 2.3.1, 2.3.0, etc and update if needed;

Search command for Unix/Linux systems:

```
grep -R 2\\. [3-4]\\. [0-2] *
```