OpenCV CI Build parameters

List of Pull requests builders

Frequently used:

Name	Description
`Docs`	Linux-based documentation builder. Additionally provides some sanity/linter checks about code
`Linux x64`	Ubuntu-based image with enabled IPPICV
`Linux OpenCL`	Ubuntu-based image with enabled IPPICV and OpenCL (Intel iGPU device)
`Linux AVX2`	Ubuntu-based image with enabled CPU_BASELINE=AVX2
`Linux x64 Debug`	Debug Ubuntu-based build with disabled SIMD intrinsics and other accelerated code paths
`Win64`	MSVS2015 with enabled IPPICV
`Win64 OpenCL`	MSVS2015 with enabled IPPICV and OpenCL (Intel iGPU device)
`Mac`	macOS X builder
`iOS`	iOS package builder (subset of architectures). Without tests launching
`Android armeabi-v7a`	Linux-based Android build for ARMv7 configuration. Without tests launching

Extra:

Name	Description
`Linux32`	32-bin Linux (64-bit host OS, but binaries in docker image are all 32-bit - no “cross-compilation”)
`Win32`	32-bit binaries (64-bit host OS)
`ARMv7`	ARMv7 + NEON. Tests are launched on NVIDIA TK-1 board
`ARMv8`	aarch64. Tests are launched on NVIDIA TX-1 board
`Android pack`	build Android package (all architectures)

Name	Description
`Custom*`	custom builders for special extra cases

Pull request common parameters

Parameters can be added into description of opened Pull request. Maintainers can adjust build parameters to extend (some special/extra cases) or reduce (documentation) scope of testing.

Note: Before merging, any reduced validation/testing scope should be normalized and re-launched builds should pass.

- `**WIP**` - marks PR as WIP. This mode allows to use some extra parameters, like tests/modules filtering. Default set of builders is not triggered in this mode.
- `force_builders=linux,Docs,Custom` - comma separated list of additional builders (both visible `Linux x64` and internal ID `linux` can be used)
- `force_builders_only=Docs` - comma separated list of builders (both visible `Linux x64` and internal ID `linux` can be used)

Pull request parameters

Some parameters are `**WIP**` protected (marked as `(WIP only)`). Without `**WIP**` flag these parameters are ignored.

Builders for Pull requests with `**WIP**` or Custom builders accept all valid parameters.

Parameters below can be applied for all builders (if allowed by `**WIP**` mode) or can be targeted for specific builder: - `parameter=value` - apply for all builders (if applicable) - `parameter:Linux x64=value` - apply for specified builder only

High-level builder configuration

- `buildworker:Custom=linux-1` (builder specific) - select assigned build worker (we using machines with different hardware/software setup).
- `build_image:Custom=ubuntu-clang:18.04` (builder specific) - select builder high-level configuration (`docker_image` is deprecated alias on this parameter)

Binaries build configuration

- `CPU_BASELINE:Custom=AVX512_SKX` (WIP only) - force CMake CPU_BASELINE option
- `CPU_DISPATCH:Linux AVX2=AVX512_SKX` (WIP only) - force CMake CPU_DISPATCH option

- `CXXFLAGS:Mac=-std=c++11` (WIP only) - set CMake C++ flags
- `CXXFLAGS_EXTRA:Custom Win=/std:c++17` (WIP only) - append CMake C++ flags
- `build_contrib=OFF` - build with/without opencv_contrib modules (speedup build)
- `build_examples=OFF` (WIP only) - disable examples build (speedup build)
- `build_shared=OFF` - select shared/static build. Static build with examples is slow in MSVS case.
- `build_world=OFF` - build opencv_world
- `build_pkgconfig=ON` - control building of pkgconfig configuration
- `build_compiler` - selects MSVC compiler (vc14 - MSVS2015, vc15 - MSVS2017, vc16 - MSVS2019)
- `build_platform` - selects MSVC platform x64, Win32, ARM

Optimizations: - `with_tbb=ON` - use TBB - `build_tbb=ON` - force building TBB from sources - `disable_ipp` - disable IPP/IPPICV

Other special parameters: - `android_pack_config:Android Pack=ndk-19.config.py` - `test_gradle:Android Pack` - force or disable Gradle builds - `build_gapi_standalone:Win64=ON` - G-API module extra testing

Binaries testing configuration

- `test_modules=dnn,python2,python3,java` (WIP only) - comma separated list of tested OpenCV modules
- `test_filter=*AlexNet*` (WIP only) - apply tests filter

Other specific cases: - `test_opencl:Custom=ON` - force testing of OpenCL - `test_bigdata:Custom=ON` (builder specific) - run huge tests - `build_parallel_tests=1` - run up to N tests in parallel

List of supported `build_image`

Linux `build_image` list

Note: - `linux-1,2,4` means list of supported workers (specify `buildworker:Custom=linux-1,linux-2,linux-4`

Basic: - `ubuntu:14.04` - `ubuntu:16.04` - `ubuntu:18.04` - `ubuntu:20.04` - `ubuntu32:16.04` (`linux-1,2,4`) - `fedora:28`, `fedora:29` (`linux-1,2,4`) - `centos:7` (`linux-1`)

Compiler: - `ubuntu-clang:18.04`

Threading: - `openmp:16.04` (`linux-1,2,4`)

HighGUI: - `qt:16.04` (`linux-1,2,4`)

Video I/O: - `ffmpeg-master` (`linux-1,2,4`) - `gstreamer:16.04`, `gstreamer:14.04` (`linux-1,2,4`)

Javascript: - `docs-js` (`linux-1,2,4` on `Docs` builder only) - `javascript` (`linux-1,2,4` on `Custom` builder only)

Android: - `android-gradle` (`linux-4` on `Android*` builders only)

DNN backends testing: - `ubuntu-openvino-2021.4.2:20.04` - `ubuntu-openvino-2020.3.0:16.04`, `ubuntu-openvino-2020.3.0:18.04` - `ubuntu-vulkan:16.04` - for testing DNN Vulkan backend - `ubuntu-cuda:18.04` (`linux-4`) - CUDA 10.0 with CUDNN? - `ubuntu-cuda11:18.04` (`linux-4`) - CUDA 11.0 with CUDNN 8

Cross-compilation for other platforms: - `powerpc64le` (`linux-1,2,4` on `Custom` builder only) - validate VSX SIMD intrinsics. Extra external buildbot for OpenCV on PowerPC - `mips64el` (`linux-1` on `Custom` builder only) - validate MIPS MSA SIMD intrinsics.

Other: - `ubuntu-cuda:16.04` (`linux-1,2,4`) - CUDA 8.0, no tests - `ubuntu-cuda:18.04` (`linux-4`) - CUDA 10.1 + CUDNN 7.6, no tests

and many deprecated/special build_images for coverage/valgrind/etc

Windows `build_image` list

- `msvs2015`, `msvs2015-win32`
- `msvs2017`, `msvs2017-win32` (`windows-1`)
- `msvs2019`, `msvs2019-win32` (`windows-1`)
- `openvino-2021.4.2` (`windows-1,3`)
- `openvino-2020.3.0` (`windows-1,2`)
- `winpack-dldt-*`, `winpack-dldt-*-debug` (`windows-1`)

Build only: - `msvs2019-ws-x64` (`windows-1`) - WindowsStore - `msvs2019-arm64` (`windows-1`) - Windows ARM64 - `cuda10`, `cuda11` (`windows-1`)

Useful extra parameters: `test_opencl=ON`

macOS X `build_image` list

- `openvino-2021.4.2`, `openvino-2020.3.0`
- `osx_framework`

Parameters for special validation cases

- SIMD optimizations validation (AVX512):

```
force_builders=Linux AVX2,Custom
buildworker:Custom=linux-3
build_image:Custom=ubuntu:18.04
CPU_BASELINE:Custom=AVX512_SKX
disable_ipp=ON
```

- SIMD optimizations validation (VSX):

```
force_builders=Linux AVX2,Custom
buildworker:Custom=linux-1,linux-2,linux-4
build_image:Custom=powerpc64le
disable_ipp=ON
```

- DNN testing (OpenVINO or new layers, tests set changes):

```
force_builders=Custom,Custom Win,Custom Mac
build_image:Custom=ubuntu-openvino-2021.4.2:20.04
build_image:Custom Win=openvino-2021.4.2
build_image:Custom Mac=openvino-2021.4.2

test_modules:Custom=dnn,python2,python3,java
test_modules:Custom Win=dnn,python2,python3,java
test_modules:Custom Mac=dnn,python2,python3,java

buildworker:Custom=linux-1
# disabled due high memory usage: test_opencl:Custom=ON
test_opencl:Custom=OFF
test_bigdata:Custom=1
test_filter:Custom=*
```

- OpenCL testing

```
force_builders=Custom,Linux AVX2,Linux OpenCL
build_image:Custom=ubuntu:18.04
buildworker:Custom=linux-5
test_opencl:Custom=ON

build_image:Linux AVX2=ubuntu:18.04
buildworker:Linux AVX2=linux-3
test_opencl:Linux AVX2=ON
```

Buildworkers special H/W capabilities

linux-1: - Intel(R) Neural Compute Stick 2 (NCS2 (USB 03e7:2485), MyriadX)

linux-2: - Intel(R) Movidius(TM) Neural Compute Stick (NCS (USB 03e7:2150), Myriad2)