Building for Apple Platforms

build_xcframework.py creates an xcframework supporting a variety of Apple platforms.

You'll need the following to run these steps: - MacOS 10.15 or later - Python 3.6 or later - CMake 3.18.5/3.19.0 or later (make sure the `cmake` command is available on your PATH) - Xcode 12.2 or later (and its command line tools)

You can then run build_xcframework.py, as below:

```
cd ~/<my_working_directory>
python opencv/platforms/apple/build_xcframework.py --out ./build_xcframework
```

Grab a coffee, because you'll be here for a while. By default this builds OpenCV for 8 architectures across 4 platforms:

- iOS (`--iphoneos_archs`): arm64, armv7
- iOS Simulator (`--iphonesimulator_archs`): x86_64, arm64
- macOS (`--macos_archs`): x86_64, arm64
- Mac Catalyst (`--catalyst_archs`): x86_64, arm64

If everything's fine, you will eventually get `opencv2.xcframework` in the output directory.

The script has some configuration options to exclude platforms and architectures you don't want to build for. Use the `--help` flag for more information.

How it Works

This script generates a fat `.framework` for each platform you specify, and stitches them together into a `.xcframework`. This file can be used to support the same architecture on different platforms, which fat `.framework`s don't allow. To build the intermediate `.framework`s, `build_xcframework.py` leverages the `build_framework.py` scripts in the ios and osx platform folders.

Passthrough Arguments

Any arguments that aren't recognized by `build_xcframework.py` will be passed to the platform-specific `build_framework.py` scripts. The `--without` flag mentioned in the examples is an example of this in action. For more info, see the `--help` info for those scripts.

Examples

You may override the defaults by specifying a value for any of the `*_archs` flags. For example, if you want to build for arm64 on every platform, you can do this:

```
python build_xcframework.py --out somedir --iphoneos_archs arm64 --iphonesimulator_archs arm
```

If you want to build only for certain platforms, you can supply the `--build_only_specified_archs` flag, which makes the script build only the archs you directly ask for. For example, to build only for Catalyst, you can do this:

```
python build_xcframework.py --out somedir --catalyst_archs x86_64,arm64 --build_only_specifi
```

You can also build without OpenCV functionality you don't need. You can do this by using the `--without` flag, which you use once per item you want to go without. For example, if you wanted to compile without `video` or `objc`, you'd can do this:

```
python build_xcframework.py --out somedir --without video --without objc
```

(if you have issues with this, try using `=`, e.g. `--without=video --without=objc`, and file an issue on GitHub.)