

OpenCV for Unity 2.6.1



System Requirements

Build Win Standalone & Preview Editor : Windows8 or later
Build Mac Standalone & Preview Editor : macOS 10.13 or later
Build Linux Standalone & Preview Editor : Ubuntu 18.04 or later
Build Android : API level 24 or later
Build iOS : iOS Version 11.0 or later

OpenCV for Unity is an Assets Plugin for using OpenCV from within Unity cross-platform game engine.

OUR ASSET FEATURES

- **Cross Platform**

Compatible with multiple platforms, allowing for app development on major platforms.

iOS & Android & Windows10 UWP support.

Win & Mac & Linux Standalone support.

WebGL support.

ChromeOS support.

- Support for **preview** in the **Editor**.
- Works with **Unity Cloud Build**.
- Support Latest Version**
Users can utilize “**OpenCV for Unity**” based on the latest version of **OpenCV (4.10.0)** for app development. (git: [opencv](#), [opencv-contrib](#))
- OpenCV Java API Supoort**
As a **clone of OpenCV Java**, which means you can use the exact **same API as OpenCV Java 4.10.0**. [OpenCV Java documentation](#)
- Easy to Use**
We provide helper functions for the interconversion of **Unity's Texture2D** and **OpenCV's Mat**. Many classes implement **IDisposable**, allowing you to manage resources using the “using” statement.
- Include Many Examples**
Includes a wide variety of example usage scenarios, which consist of **scene files** and **script codes**. By running these sample applications, **you can learn how to develop OpenCV applications effectively**.

[OpenCVForUnity Examples \(GitHub\)](#)

[EnoxSoftware repositories \(GitHub\)](#)
- AR VR MR**
Can be utilized for developing applications using **Augmented Reality**, **Virtual Reality**, and **Mixed Reality** technologies.
- Deep Learning**
Provides support for the **dnn module**, including various frameworks such as **ONNX**, **TensorFlow**, **caffe**, **Torch**, **Darknet**, and more. See [OpenCV wiki \(GitHub\)](#)
On all platforms, inference in the Dnn module uses the **CPU backend** by default; only Windows platforms can use the **CUDA backend** by following [additional steps](#).
- Use of WebCamTexture**
Supports input from **Unity's WebCamTexture**, allowing you to perform **real-time image processing** on camera footage.
- Works with many hardware**
Compatible with a **wide range of hardware gadgets** beyond just PCs and smartphones, allowing it to run on various devices.
(e.g. **HoloLens1 / Hololens2**, **Nreal Light**, **Oculus**, **Kinect**, **RealSense**, **ZED 2 or ZED Mini stereo camera**, and **Raspberry Pi**).
- Visual Scripting Support**
By utilizing the **VisualScripting With OpenCVForUnity Example**, you can leverage all the methods available in OpenCVforUnity within the **Unity's Visual Scripting development environment**.

[VisualScripting With OpenCVForUnity Example \(GitHub\)](#)

[Official Site](#) | [ExampleCode](#) | [Android Demo](#) [WebGL Demo](#) | [Product Introduction](#)
[Video Setup Guide Video](#) | [Forum](#) | [API Reference](#) | [Support Modules](#)

Please refer to [OpenCV official document](#) for the details of the argument of the method.

OpenCV for Unity uses **OpenCV** under **Apache 2 license**; see `Notices.txt` file in package for details.

Example code using OpenCV for Unity is available.

- [MarkerBased AR Example](#)
- [MarkerLess AR Example](#)
- [FaceTracker Example](#)
- [FaceSwapper Example](#)
- [FaceMask Example](#)
- [RealTime FaceRecognition Example](#)
- [Vuforia with OpenCV for Unity Example](#)
- [AVPro with OpenCV for Unity Example](#)
- [Kinect with OpenCV for Unity Example](#)
- [HoloLens with OpenCV for Unity Example](#)
- [NatDevice with OpenCVForUnity Example](#)
- [NatCorder with OpenCVForUnity Example](#)
- [VisualScriptingWithOpenCVForUnityExample](#)
- [ARFoundationWithOpenCVForUnityExample](#)
- [NrealLight with OpenCVForUnity Example](#)
- [FfmpegWithOpenCVForUnityExample](#)
- [VideoPlayerWithOpenCVForUnityExample](#)
- [NativeGalleryWithOpenCVForUnityExample](#)

Version changes :

2.6.1 [Common]Updated to OpenCV4.10.0. [Common]Added MultiSource2MatHelperExample.

2.6.0 [iOS]Added separate plugin files for iOS for devices and simulators. [WebGL]Added plugin files with only simd enabled.

2.5.9 [Common]Added PoseSkeletonVisualizer to HandPoseEstimationMediaPipeExample and PoseEstimationMediaPipeExample. [Common]Changed to use unsafe code by default. [Common]Optimized the amount of memory allocation, mainly in the Convertor class.

2.5.8 [Common]Updated to OpenCV4.9.0. [Common]Added DebugMatUtilsExample and MultiObjectTrackingExample. [Common]Updated VideoWriterExample, VideoWriterAsyncExample, TextRecognitionCRNNExample, TextRecognitionCRNNWebCamExample and TrackingExample. [Common]Changed the minimum supported version to Unity2020.3.48f1. [WebGL]Added support for "WebAssembly 2023". [iOS]Changed "Target minimum iOS Version" to 11.0. [Common]Removed TrackerGOTURN from TrackingExample and added TrackerVit.

2.5.7 [Common]Added FeatureMatchingExample. [WebGL]Added a plugin file with threads and simd enabled for the WebGL platform. This update removes support for the WebGL platform in Unity 2021.1 and below. (Select MenuItem[Tools/OpenCV for Unity/Open Setup Tools/WebGL Settings])

2.5.6 [Common]Added PoseEstimationMediaPipeExample. [Common]Updated VideoWriterExample, VideoWriterAsyncExample (support URP and HDRP) .

2.5.5 [Common]Updated to OpenCV4.8.0. [Common]Added ImageCorrectionExample, YuNetFaceDetectionExample and FaceDetectionYuNetV2Example. [Common]Updated HumanSegmentationExample, BarcodeDetectorExample. [Windows]Added Support for ARM64. [WebGL]Added Unity2023.2 or later support.

2.5.4 [Common]Added VideoWriterAsyncExample. [Common]Updated SimpleBlobExample, HumanSegmentationExample and HandPoseEstimationExample.

2.5.3 [Common]Added HandPoseEstimationExample and FacialExpressionRecognitionExample.

2.5.2 [Common]Added YOLOv4ObjectDetectionExample, YOLOv7ObjectDetectionExample, YOLOXObjectDetectionExample and NanoDetPlusObjectDetectionExample. [Common]Updated HandPoseEstimationExample, ArUcoExample, ArUcoWebCamExample and ArUcoCameraCalibrationExample.

2.5.1 [Common]Updated to OpenCV4.7.0. [Common]Added ImageClassificationMobilenetExample and HandPoseEstimationExample. [Common]Added TrackerNano to TrackingExample. [Lumin]Removed Lumin platform support (for MagicLeapOne). [Common]Add a button to SetupTools to automatically add scenes under the "Examples" folder to "Scenes In Build".

2.5.0 [Common]Added TransformECCEExample, HumanSegmentationExample and ImageClassificationPPResnetExample. [Common]Update TextOCRWebCamExample. [Common]Changed the setup procedure to use the SetupToolsWindow. [Common]Change the namespace under "OpenCVForUnity/Editor" folder from "OpenCVForUnity" to "OpenCVForUnity.Editor". [Common]Added the ExampleAssetsDownloaderWindow that automatically downloads the necessary files to Examples. [Common]Added "OpenCVForUnity" folder under "StreamingAssets" folder. [Common]Added function to automatically move the StreamingAssets folder. [WebGL]Added Unity2022.2 or later support.

2.4.9 [Common]Added LegacyTrackingExample and LightweightPoseEstimationWebCamExample. [Common]Renamed "WebCamTextureExample" to "WebCamExample". [WebGL]Fixed plugins for the WebGL platform.

2.4.8 [Common]Updated to OpenCV4.6.0. [Common]Added KNNExample, PhysicalGreenScreenExample and LightweightPoseEstimationExample. [Common]Update VideoCaptureToMatHelper, GreenScreenExample and QRCodeDetectorExample.

2.4.7 [Common]Updated to OpenCV4.5.5. [Common]Added FaceDetectorYNWebCamTextureExample, FaceRecognizerSFExample and QRCodeEncoderExample.

2.4.6 [Common]Updated to OpenCV4.5.4. [Android]Added Support for ChromeOS (x86 and x86_64 architectures).

2.4.5 [Common]Updated to OpenCV4.5.3. [Common]Added BarcodeDetectorExample and BarcodeDetectorWebCamTextureExample.

2.4.4 [Common]Updated to OpenCV4.5.2. [Common]Added VideoCaptureCameraInputExample and BackgroundSubtractorComparisonExample. [Common]Updated TrackingExample.

2.4.3 [Common]Added a downloader script to automatically set up the Dnn module example. [WebGL]Added exclude_contrib version for build size reduction.

2.4.2 [Common]Added Assembly Definitions. [Common]Fixed LibFaceDetectionV3Example.

2.4.1 [Common]Updated to OpenCV4.5.0. [Common]Added DaSiamRPNTrackerExample.

2.4.0 [Common]Updated to OpenCV4.4.0. [Common]Added TextOCRExample.

[Common]Updated YoloObjectDetectionExample (Yolo v4).

2.3.9 [Common]Updated to OpenCV4.3.0. [Common]Added LibFaceDetectionV2Example, LibFaceDetectionV3Example, ColorizationExample and DocumentScannerExample. [Common]Update ArUcoCameraCalibrationExample and WrapPerspectiveExample.

2.3.8 [Common]Updated to OpenCV4.2.0. [UWP]Added ARM64 Architecture.

[WebGL]Added opencvforunity.bc with multi-threading enabled. [Common]Added FastNeuralStyleTransferExample and LibFaceDetectionExample. [Common]Added MatIndexer class and MatUtils class. [Common]Update ComicFilterExample, VideoCaptureExample, OpenPoseExample and MatBasicProcessingExample.

2.3.7 [WebGL]Fixed build errors that occur when DevelopmentBuild is enabled on the WebGL platform. [Common]Added optimization code using NativeArray class. (require PlayerSettings.allowUnsafeCode flag, “OPENCV_USE_UNSAFE_CODE” ScriptingDefineSymbol and Unity2018.2 or later.) [iOS]Fixed build errors that occur on the iOS platform with Unity2019.3 or later. [Common]Updated to WebCamTextureToMatHelper.cs v1.1.1.

2.3.6 [WebGL]Fixed “Plugins/WebGL/2018.2/opencvforunity.bc”. [Common]Added multi-dimensional Mat example to MatBasicProcessingExample. [Common]Fixed ARUtils.cs.

2.3.5 [Common]Updated to OpenCV4.1.0. [Windows, Android]Added dynamic link library version.

2.3.4 [Common]Added MaskRCNNEExample. [WebGL]Added Unity2019.1 or later support.

2.3.3 [Common]Updated to OpenCV4.0.0. [Common]Re-assigned namespace as was classified by module names. [Common]Updated to WebCamTextureToMatHelper.cs v1.1.0. [Common]Updated to ImageOptimizationHelper v1.1.0 [Common] Added Utils_GetFilePathExample, FaceMarkExample and QRCodeDetectorExample.

2.3.2 [macOS]Removed 32bit architecture(i386) from opencvforunity.bundle.

2.3.1 [Common]Updated to OpenCV3.4.2. [Android,UWP]Fixed Utils.setDebugMode() method on the IL2CPP backend. [Common]Added DnnObjectDetectionExample and DnnObjectDetectionWebCamTextureExample.

2.3.0 [iOS]Added a function to automatically remove the simulator architecture(i386,x86_64) at build time. [Common] Improved OpenCVForUnityMenuItem.setPluginImportSettings() method.

2.2.9 [Linux]Simplified the Linux platform setup procedure. [Common]Added support for Utils. setDebugMode() method on all platforms. [Common]Updated to WebCamTextureToMatHelper.cs v1.0.9. [Common]Added MatToTextureInRenderThreadExample and AlphaBlendingExample.

2.2.8 [Common]Updated to WebCamTextureToMatHelper.cs v1.0.7. [Common]Added MatBasicProcessingExample. [Common]Fixed WebCamTextureToMatExample, WebCamTextureToMatHelperExample, ArUcoExample. [Common]Added flip flag to Utils.fastMatToTexture2D() method and Utils.fastTexture2DToMat() method. [Common]Added throwException flag to Utils.setDebugMode() method.

2.2.7 [Common]Updated to OpenCV3.4.1. [Common]Added OpenPoseExample(The model file is not included in this asset.), KalmanFilterExample, ArUcoCameraCalibrationExample. [Common]Fixed VideoWriterExample, VideoCaptureExample, ImwriteScreenCaptureExample, CamShiftExample, TrackingExample, HandPoseEstimationExample, ArUcoCreateMarkerExample, ArUcoExample, ArUcoWebCamTextureExample. [Common] Updated to WebCamTextureToMatHelper.cs v1.0.6.

2.2.6 [Android]Added arm64-v8a Architecture. [Common]Added ImwriteScreenCaptureExample.

2.2.5 [Common] Updated to WebCamTextureToMatHelper.cs v1.0.4. [Common] Fixed MobileNetSSDExample and MobileNetSSDWebCamTextureExample.

2.2.4 [Common]Updated to OpenCV3.3.1. [Common]Added ResnetSSDFaceDetectionExample, YoloObjectDetectionExample, YoloObjectDetectionWebCamTextureExample.

2.2.3 [Common]Updated to WebCamTextureToMatHelper.cs v1.0.3. [iOS]

opencv2.framework is changed from static framework to embededd framework. (Target minimum iOS Version must be set to 8.0 or higher.)

2.2.2 [Common]Added TextRecognitionExample.

2.2.1 [Common]Updated to OpenCV3.3.0. [Common]Added dnn module.(win,mac,ios,android platform) [Common]Added img_hash, reg, text module.(all platform) [Common]Added MobileNetSSDExample, MobileNetSSDWebCamTextureExample, TensorFlowWebCamTextureExample, ThinPlateSplineShapeTransformerExample, TextDetectionExample, VideoWriterExample. [Common]WindowsStoreApp8.1 & WindowsPhone8.1 support have been deprecated.

2.2.0 [Common]Updated WebCamTextureToMatHelper.cs v1.0.2 [Common]Improved Utils.getFilePathAsync().

2.1.9 [WebGL]Fixed Utils.getFilePathAsync() method.

2.1.8 [Common]Added PCAExample. [Common]Updated WebCamTextureToMatHelper.cs and OptimizationWebCamTextureToMatHelper.cs(Changed several method names.).

2.1.7 [Common]Improved Utils.getFilePath() and Utils.getFilePathAsync().

[Common]Improved WebCamTextureAsyncDetectFaceExample.cs. [Common] Fixed the const value of Calib3d class.

2.1.6 [Common]Fixed fastMatToTexture2D() method.

2.1.5 [Common]Updated to OpenCV3.2.0. [Common]Added fuzzy, phase_unwrapping, saliency, shape, tracking module. [Common]Added TrackingSample. [iOS]Added ios_exclude_contrib.zip for build size reduction. [Android]Added android_exclude_contrib.zip for build size reduction.

2.1.4 [Common]Changed the scene name.("Sample" to "Example") [Common]Fixed ArUcoTexture2DExample and ArUcoWebCamTextureExample. [Common]Added ConnectedComponentsExample. [Common]Added GreenScreenExample.

2.1.3 [UWP]Added OpenCVForUnityUWP_Beta3.zip.

2.1.2 [Common]Fixed WebCamTextureToMatHelper.cs.(flipVertical and flipHorizontal flag)

2.1.1 [Common]Fixed OpenCVForUnityMenuItem.cs.(No valid name for platform: 11 Error) [Common]Added Utils.textureToTexture2D() method. [Common]Added Mat class operators. [Common]Added PolygonFilterSample.

2.1.0 [Common]Fixed WebCamTextureToMatHelper class. [Common]Added Utils.getVersion(). [Common]Fixed Utils.getFilePathAsync().

2.0.9 [WebGL]Added WebGL(beta) support.(Unity5.3 or later)

2.0.8 [Common]Improved WebCamTextureHelper class. [Common]Fixed ArUcoSample.

2.0.7 [Common]Added aruco, structured_light, xfeatures2d module. [Common]Added ArUcoSample, GrabCutSample, InpaintSample, MatchShapesSample, MSERSample.

2.0.6 [WSA]Fixed an issue where Windows App Certification Kit fails.

2.0.5 [Common]Added HOGDescriptorSample.

2.0.4 [Android]Added Support for Split Application Binary (.OBB) [Android]Removed opencvforunity.jar.

2.0.3 [Common]Added SVMSample. [Common]Fixed VideoCaptureSample and WebCamTextureAsyncDetectFaceSample. [UWP]Added OpenCVForUnityUWP_Beta2.zip.

2.0.2 [Common]Fixed CS0618 warnings: `UnityEngine.Application.LoadLevel(string)' is obsolete: `Use SceneManager.LoadScene'.

2.0.1 [OSX]Fixed SIGILL Exception. [Common]Added Utils.setDebugMode() method. [Common]Added MatchTemplateSample, StereoBMSample, SeamlessCloneSample and WebCamTextureDetectCirclesSample. [Common]Added flipVertical flag, flapHorizontal flag and GetWebCamDevice() method to WebCamTextureToMatHelper.cs.

2.0.0 [Common]Updated to OpenCV3.1.0. [Common]Included Old Version based on "OpenCV2.4.11". [Common] Included Beta Version of Windows10 UWP Support.(This is

beta version based on OpenCV3.0.0. opencv_contrib modules is not supported.)

Beta16 [iOS]Fixed libopencvforunity.a Bitcode Setting.

Beta15 [Common]Fixed WebCamTextureToMatHelper.cs.(Add didUpdateThisFrame () method)

Beta14 [Common]Fixed WebCamTextureToMatHelper.cs.(Bug of rotation conversion from WebCamTexture to Mat in Win,Mac StandAlone Build)

Beta13 [Common]Added fastTexture2DToMat() and fastMatToTexture2D(). [Common] Renewed the samples using WebCamTextureToMatHelper.(Supports all screen orientation.)

Beta12 [iOS]Fixed malloc_error that occurs in Unity5.3.1p2.

Beta11 [iOS]Enabled Jpeg format.(Added mjpeg format support in VideoCapture class)

Beta10 [iOS]Enabled Bitcode.

Beta9 [UWP]Added support for Windows10 UWP.(This is a test version. opencv_contrib modules is not supported.)

Beta8 [Common]Fix FaceRecognizerSample. [Common] Delete the method using Default parameter specifiers. [Android] Compile the library using “armabi-v7a with NEON” option.

Beta7 [Common]Add WrapPerspectiveSample, HandPoseEstimationSample.

Beta6 [iOS]Fix WebCamTexture bug of SampleScene in Unity5.2.

Beta5 [Linux]Add Linux Support. [WindowsStoreApp8.1]Support for methods using Low-level Native Plugin Interface. [Common]Rewrite SampleScene.

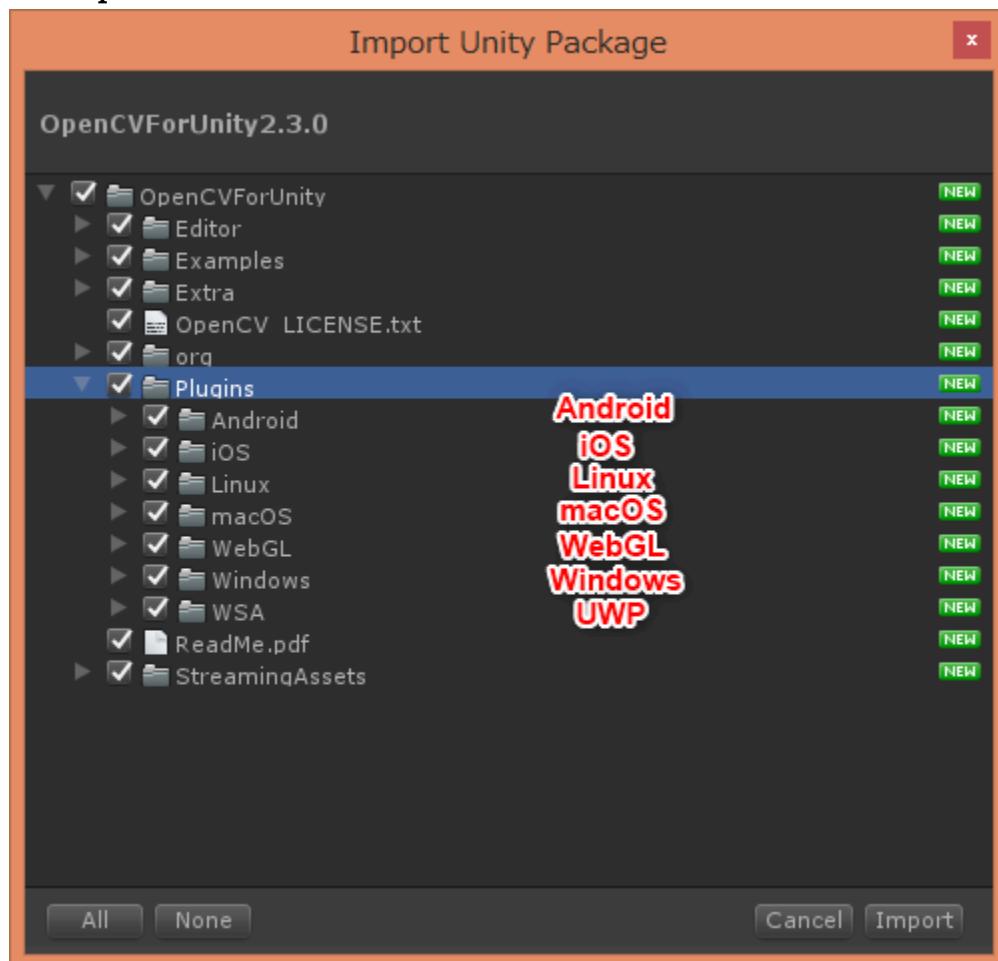
Beta4 [Common]Add Utils. getGraphicsDeviceType(). [Common]Add SampleScene Setup Tutorial Video for Unity5.

Beta3 [Common]Add CamShiftSample.(Object Tracking) [Common]Add OpenCVForUnityMenuItem.cs.(This script set plugin import settings automatically from MenuItem.)

Beta2 [iOS] Fix problem when working with Metaio(UnityAppController problem). [Common]Add [System.Serializable] to basic class. [Common] change folder name from “OpenCVForUnity/OpenCVForUnity_Editor/” to “OpenCVForUnity/Editor/”. [iOS]Move “OpenCVForUnity/OpenCVForUnity_Editor/opencv2.framework” to “OpenCVForUnity/Plugins/iOS”folder.

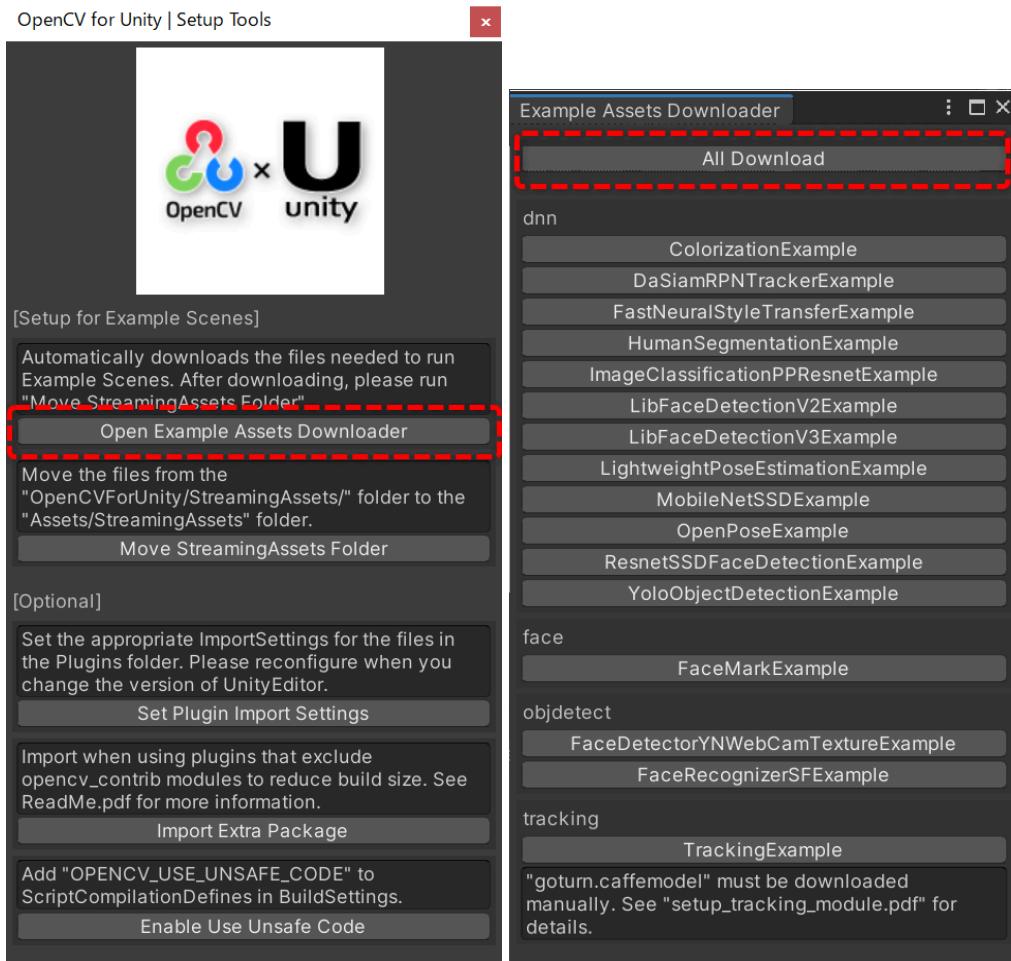
Quick setup procedure to get started with development ([Setup Guide Video](#)) :

1. Import the OpenCVForUnity.package. You do not need to import plug-in files for platforms not supported by your project. **If there is a previous version of OpenCVForUnity in the project, please delete the OpenCVForUnity folder first and then import the new version.**

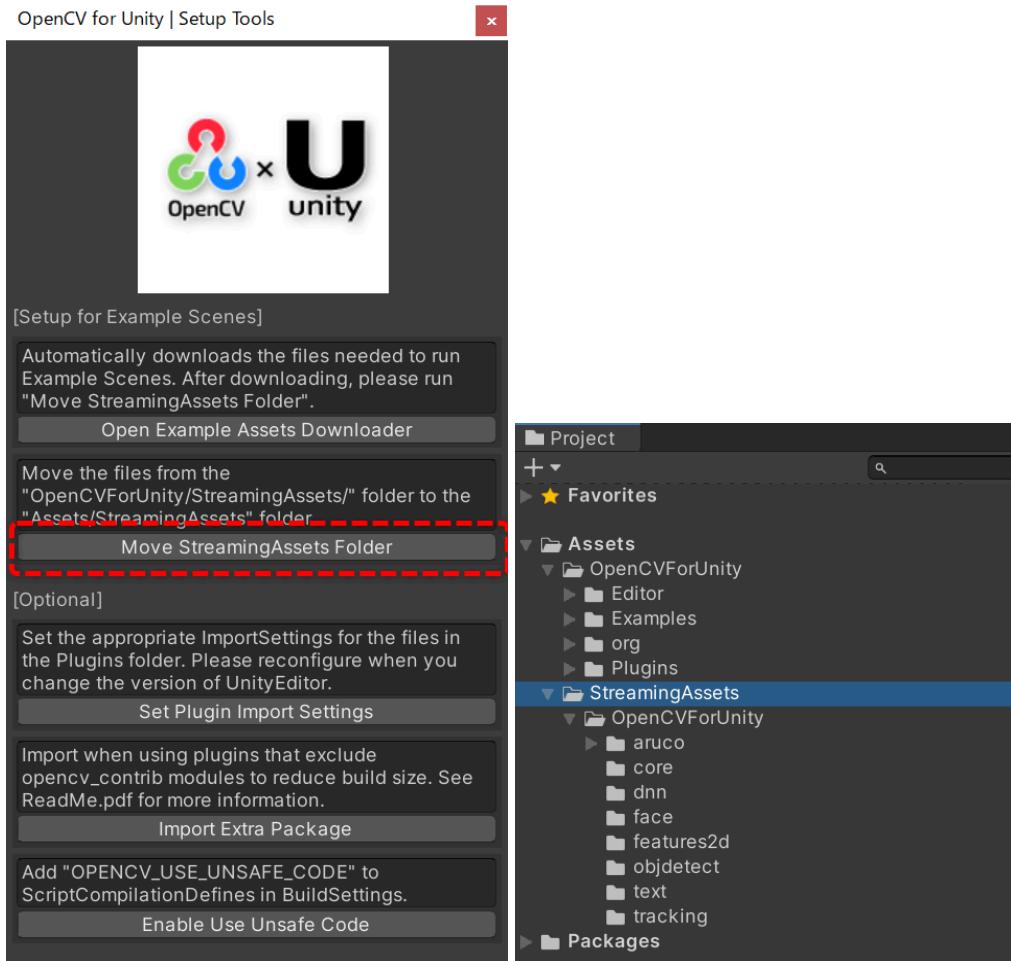


Quick setup procedure to run the example scenes ([Setup Guide Video](#)) :

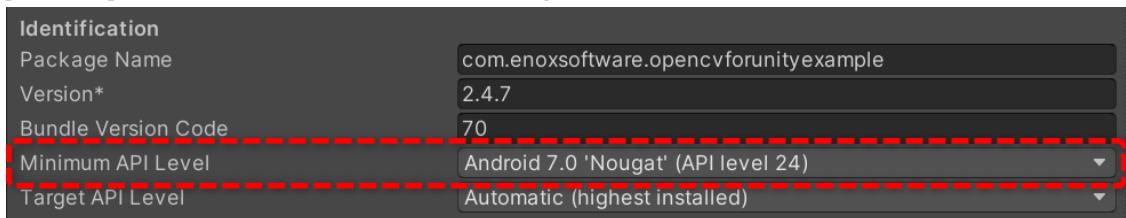
1. Import the OpenCVForUnity.package.
2. Select MenuItem[Tools/OpenCV for Unity/Open Setup Tools].
3. Click the [Open Example Download] button, then click the [All Download] button.



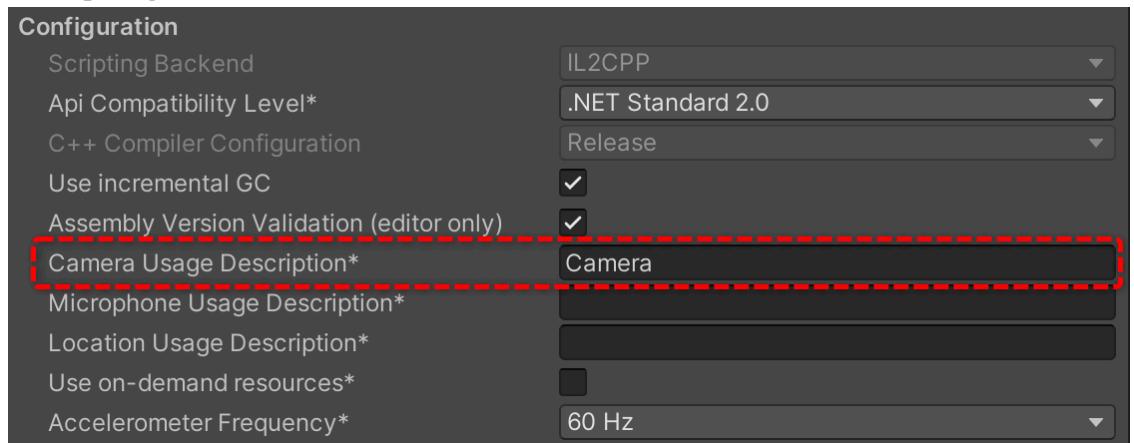
4. Click the [Move StreamingAssets Folder] button.



5. [Android] Set Minimum API Level to 24 or higher.



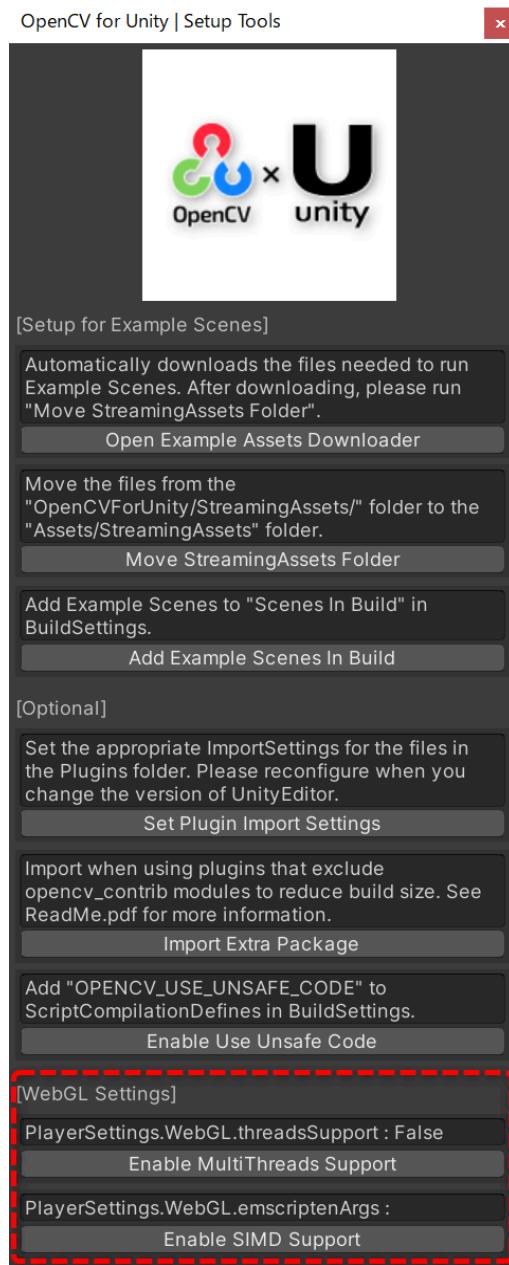
6. [iOS] Set [PlayerSettings]-[Other Settings]-[Configuration]-[Camera Usage Description].



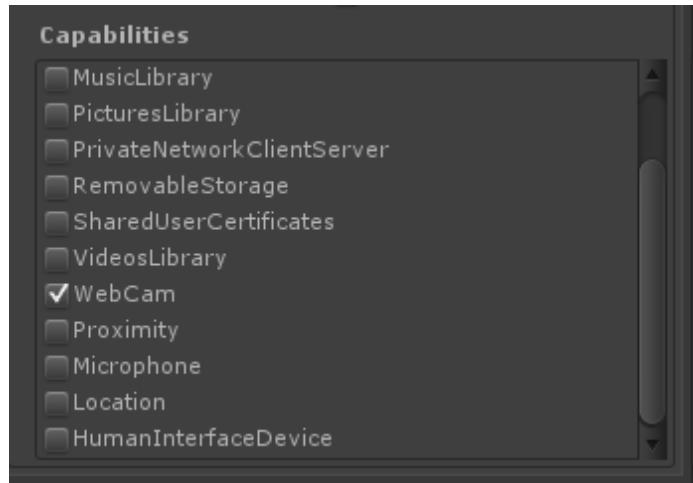
Set Target minimum iOS Version to 11.0 or higher.



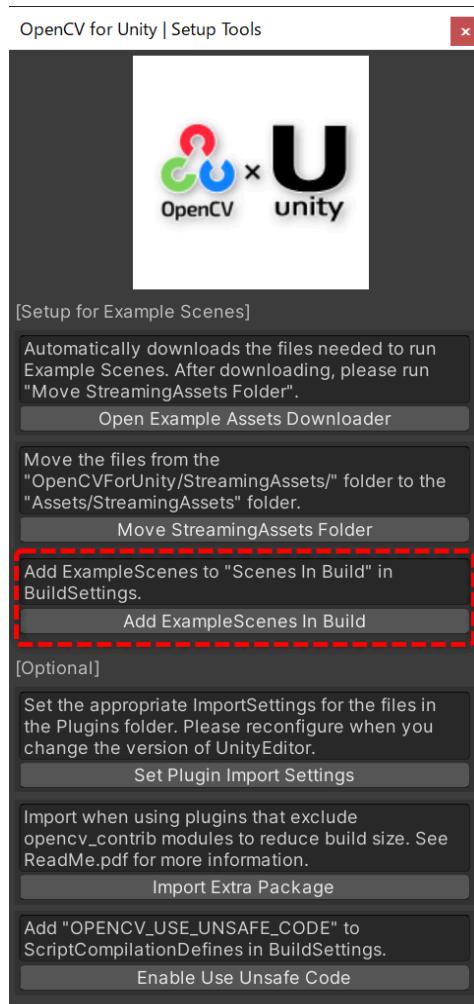
7. [WebGL] If you want to enable threads or SIMD optimization, click the [**Enable MultiThreads Support**] button or the [**Enable SIMD Support**] button. (The executing browser must support threading and SIMD.)

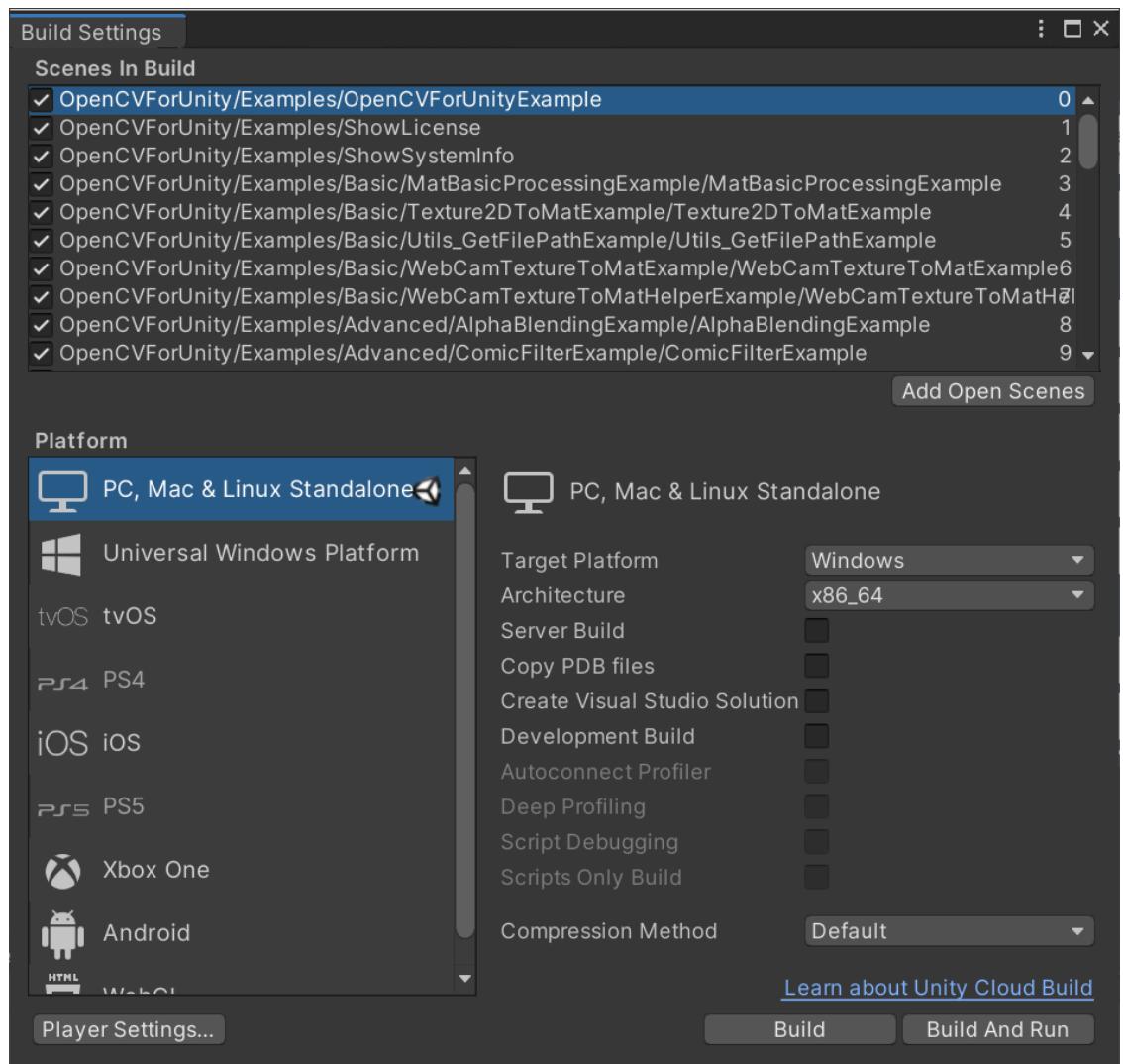


8. [Windows10 UWP] If use webCamTextue class, Please choose “WebCam” in [PlayerSettings]-[PublishingSettings]-[Capabilities].

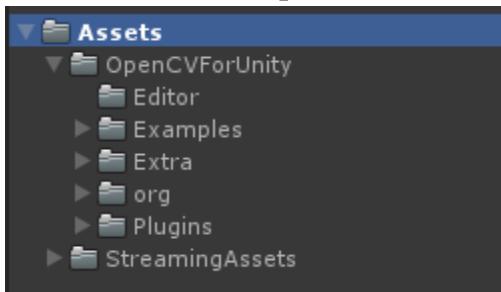


9. Add all of the “***.unity” in the “OpenCVForUnity/Examples” folder to [Build Settings] – [Scene In Build].



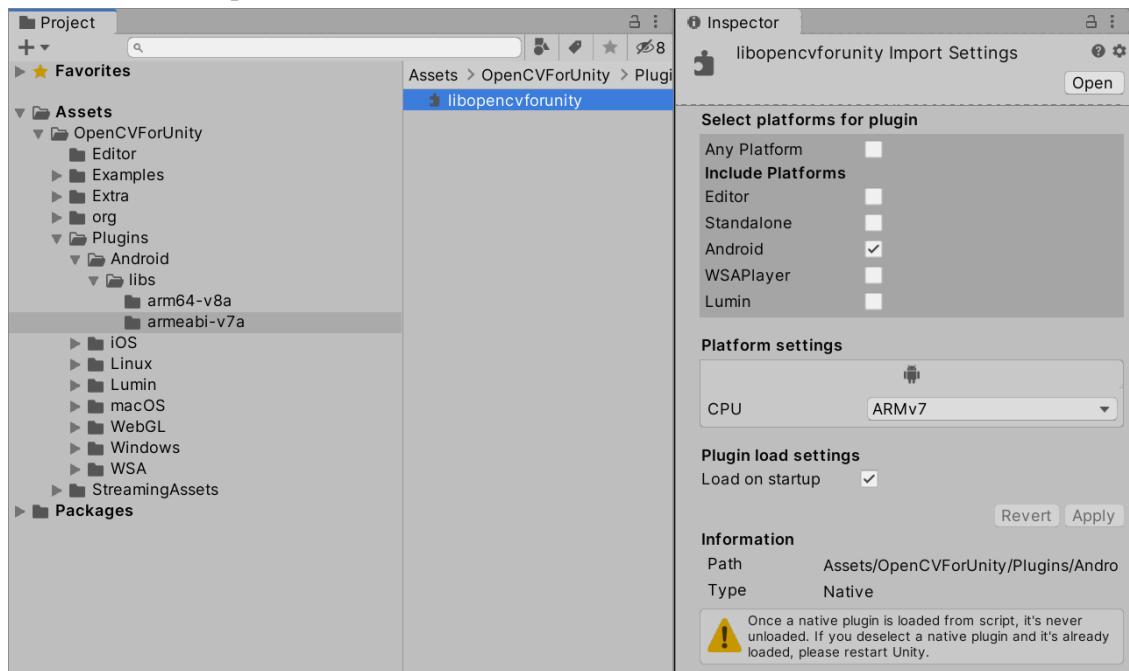


Screenshot after the setup

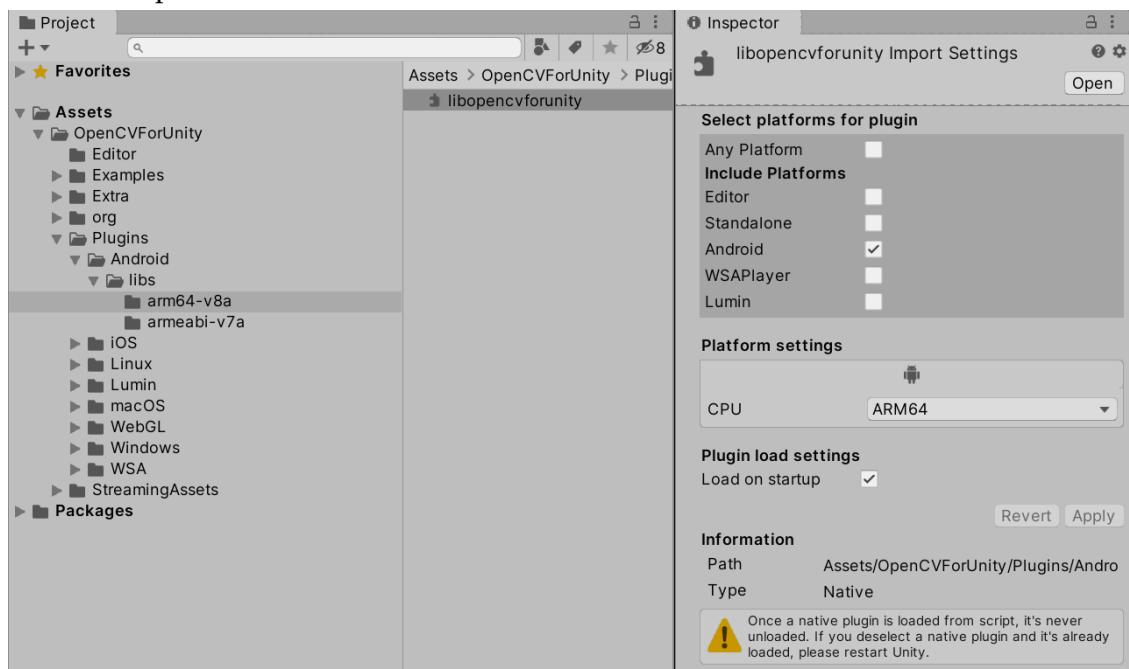


Android Setup Procedure :

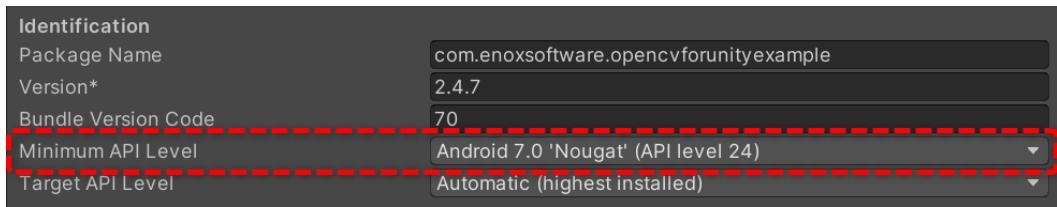
- “OpenCVForUnity/Plugins/libs/armeabi-v7a/*.so” – Select platform Android and CPU ARMv7 in Inspector.



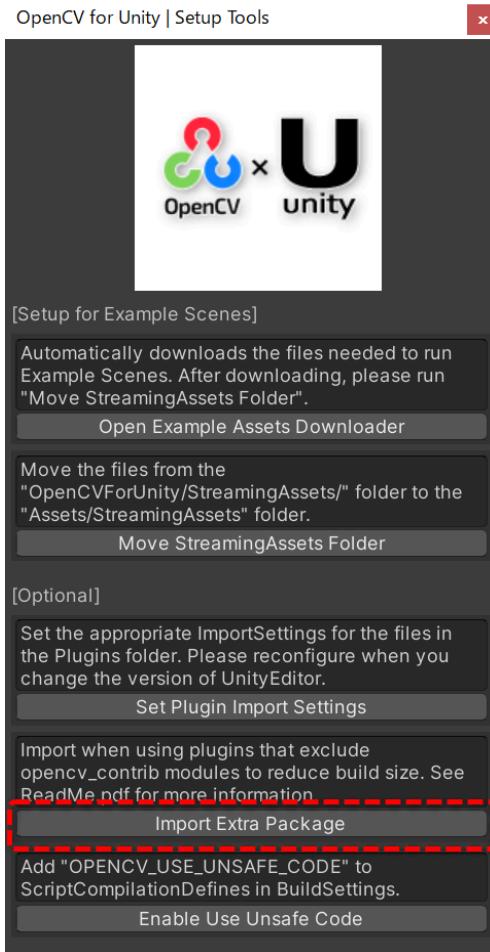
- “OpenCVForUnity/Plugins/libs/arm64/*.so” – Select platform Android and CPU arm64 in Inspector.



- Set Minimum API Level to 24 or higher.



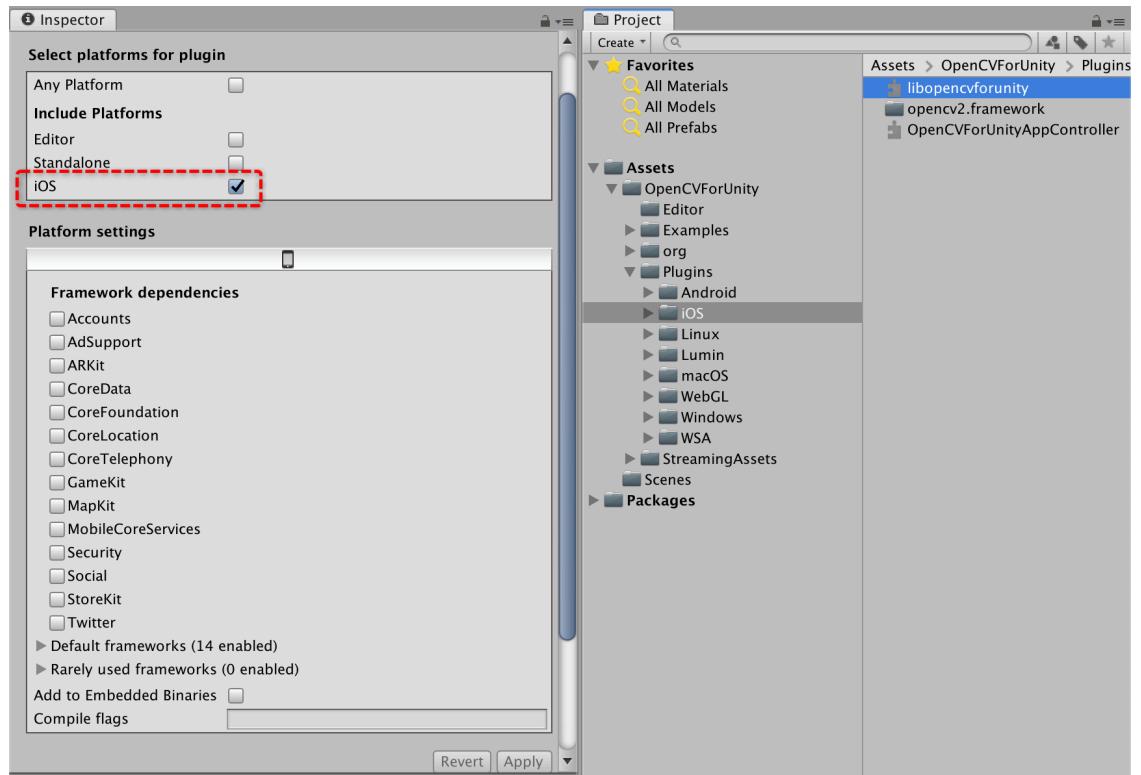
- If you do not use opencv_contrib module, build size will be reduced by using native plugin file excluding opencv_contrib module.



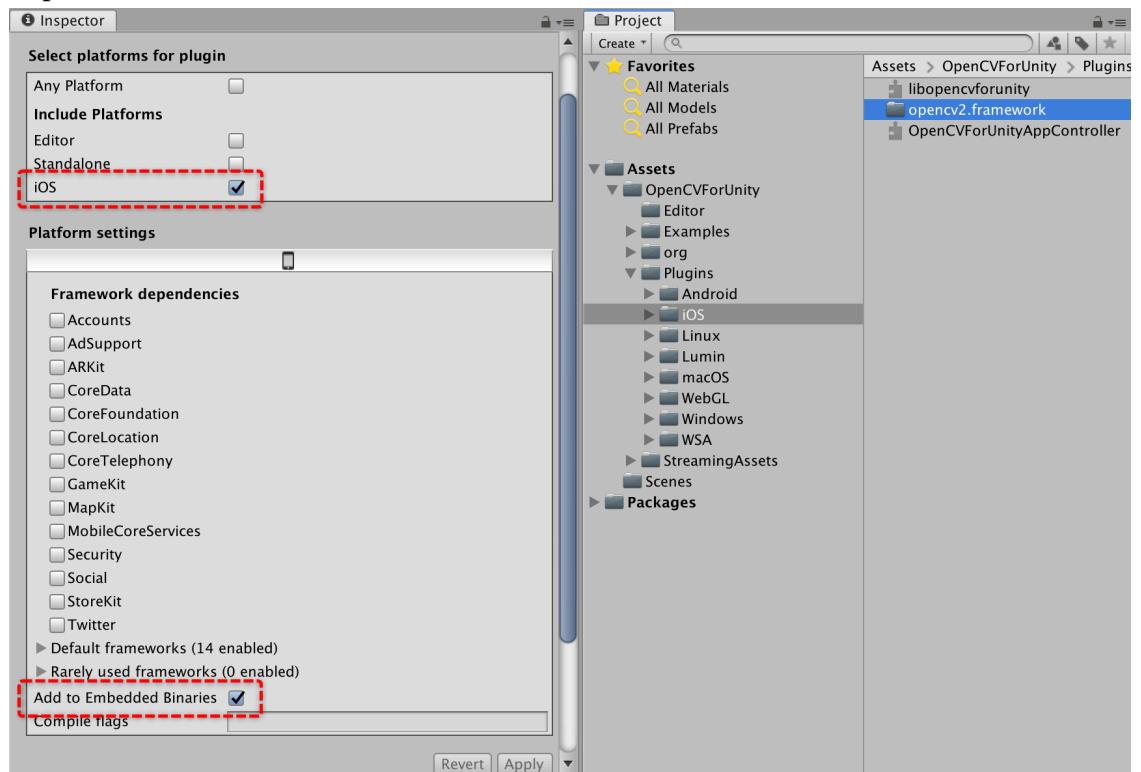
1. Replace “OpenCVForUnity/Plugins/Android/libs” folder to “OpenCVForUnity/Extra/exclude_contrib/Android/libs” folder.
2. Click the [Set Plugin Import Settings] button.
3. Delete “OpenCVForUnity/Assets/OpenCVForUnity/org/opencv_contrib” folder and “OpenCVForUnity/Examples/ContribModules” folder.

iOS Setup Procedure :

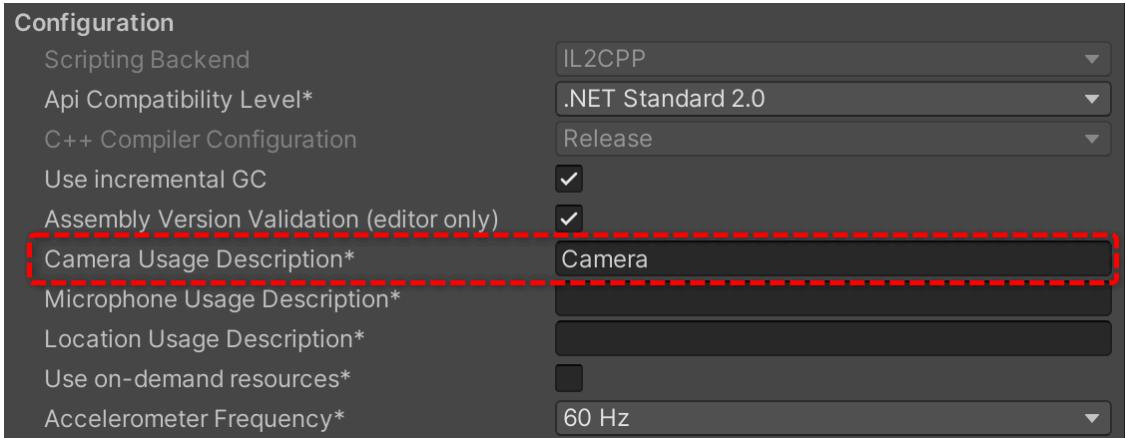
- “OpenCVForUnity/Plugins/iOS/libopencvforunity.a” – Select platform iOS in Inspector.



- “OpenCVForUnity/Plugins/iOS/opencv2.framework” – Select platform iOS in Inspector.



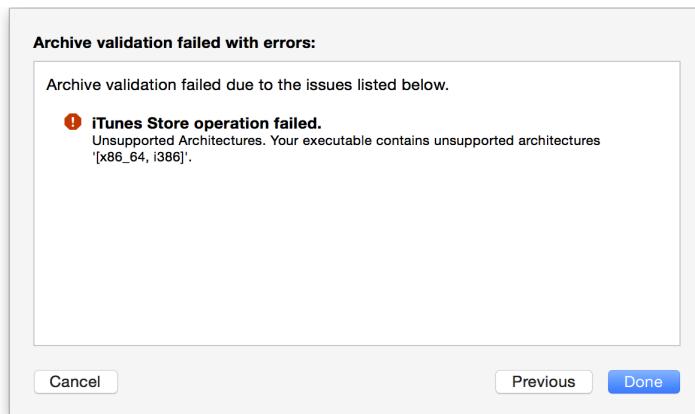
- If iOS platform, Set [PlayerSettings]-[Other Settings]-[Configuration]-[Camera Usage Description].



- Set Target minimum iOS Version to 11.0 or higher.

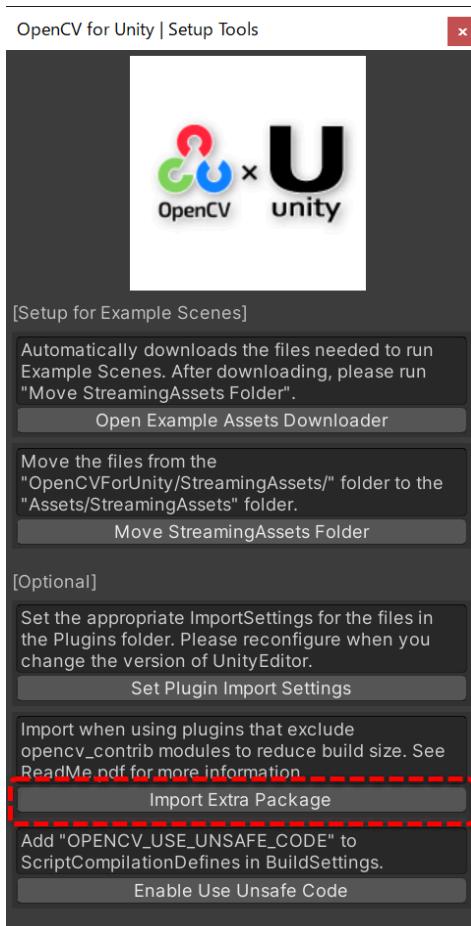


- When exporting ipa file, you need to remove the unneeded architectures from opencv2.framework, before submitting it.



Please see Q & A No.9 for details.

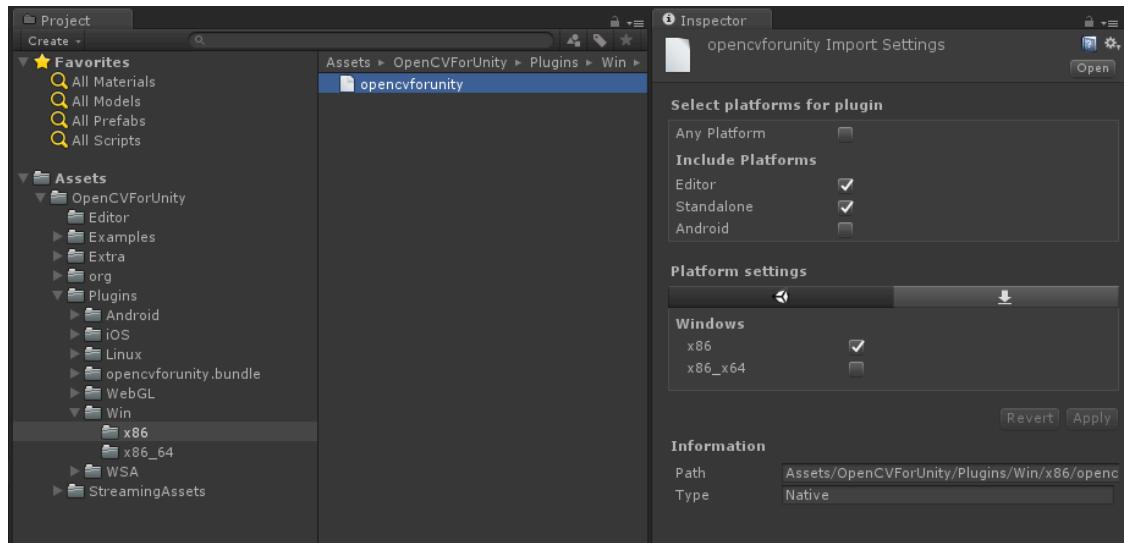
- If you do not use opencv_contrib module, build size will be reduced by using native plugin file excluding opencv_contrib module.



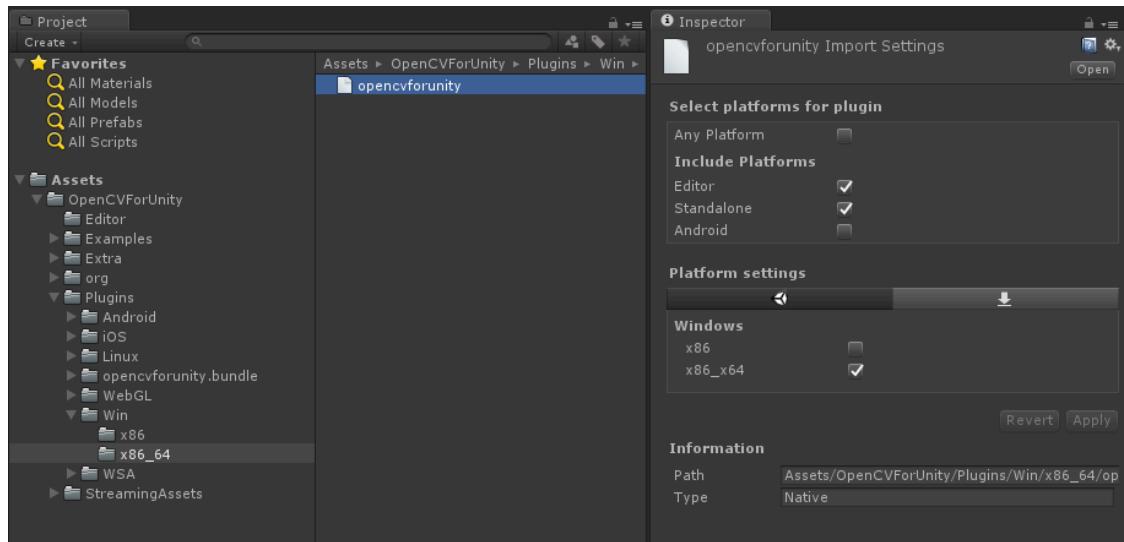
1. Replace “OpenCVForUnity/Plugins/iOS” folder to “OpenCVForUnity/Extra/exclude_contrib/iOS” folder.
2. Click the [Set Plugin Import Settings] button.
3. Delete “OpenCVForUnity/Assets/OpenCVForUnity/org/opencv_contrib” folder and “OpenCVForUnity/Examples/ContribModules” folder.

Windows Standalone Setup Procedure :

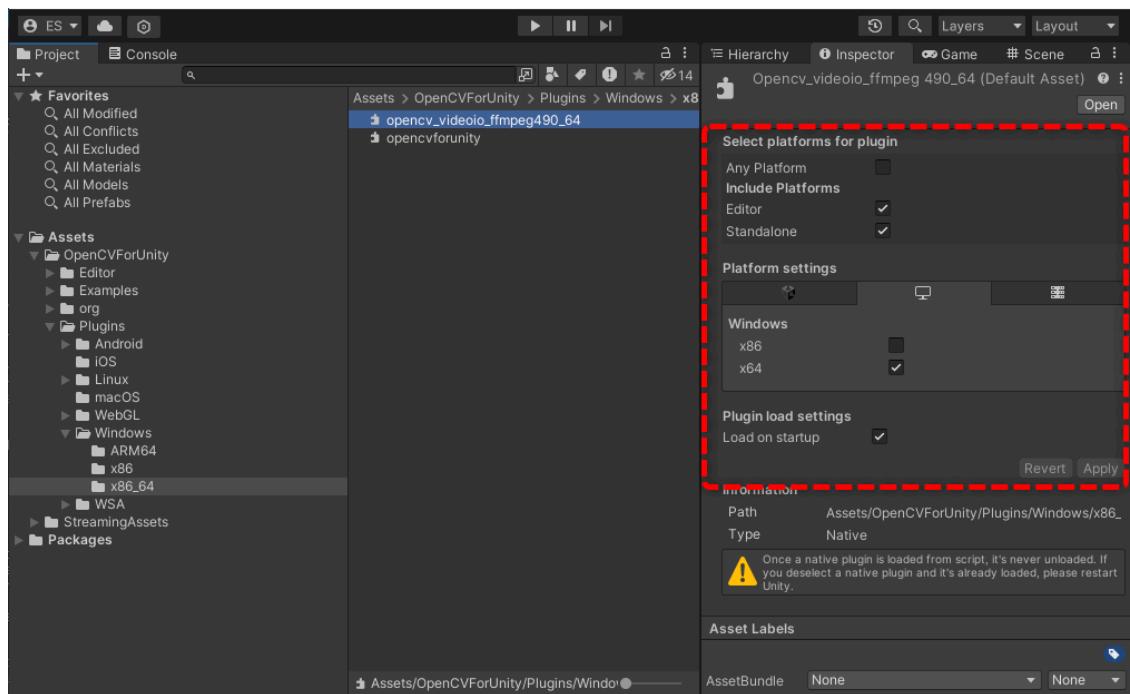
- “OpenCVForUnity/Plugins/Windows/x86/opencvforunity.dll” – Select platform Editor,Standalone and CPU x86 and OS Windows in Inspector.



- “OpenCVForUnity/Plugins/Windows/x86_64/opencvforunity.dll” – Select platform Editor,Standalone and CPU x86_64 and OS Windows in Inspector.



- If you want to use more video formats with the "Video Capture (string filename)" or "VideoWriter" method, setup is required.
 1. Download "OpenCV for Windows Version 4.10.0" (<http://opencv.org/downloads.html>).
 2. Copy "opencv/build/bin/opencv_videoio_ffmpeg490_64.dll" to "OpenCVForUnity/Plugins" Folder.
 3. Setup "opencv_videoio_ffmpeg490_64.dll" Inspector Settings.



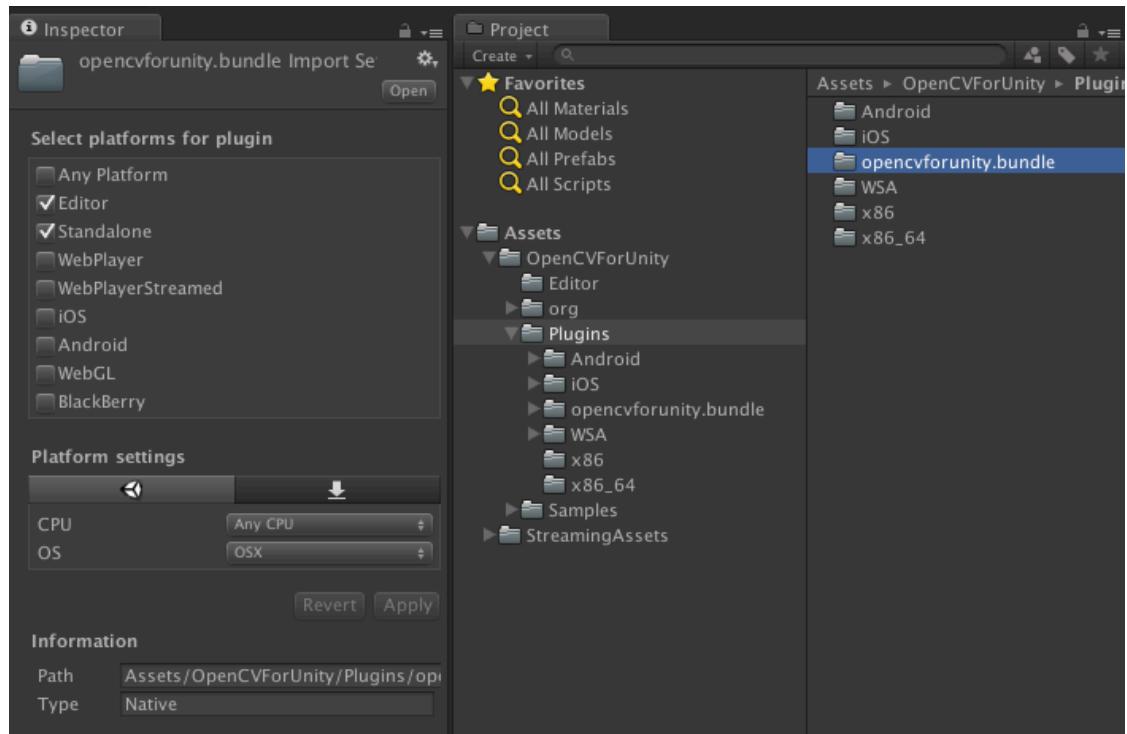
You can use ffmpeg for the backend by setting `Videoio.CAP_FFMPEG` in the `apiPreference` as shown in the code below. The ffmpeg backend may allow you to specify more file formats than the default backend.

```
capture.open(Utils.getFilePath(VIDEO_FILENAME), Videoio.CAP_FFMPEG);
Debug.Log("capture.getBackendName(): " + capture.getBackendName());
```

```
writer.open(savePath, Videoio.CAP_FFMPEG, VideoWriter.fourcc('M', 'J', 'P', 'G'), 30, new
Size((int)captureRectPixel.width, (int)captureRectPixel.height));
Debug.Log("writer.getBackendName(): " + writer.getBackendName());
```

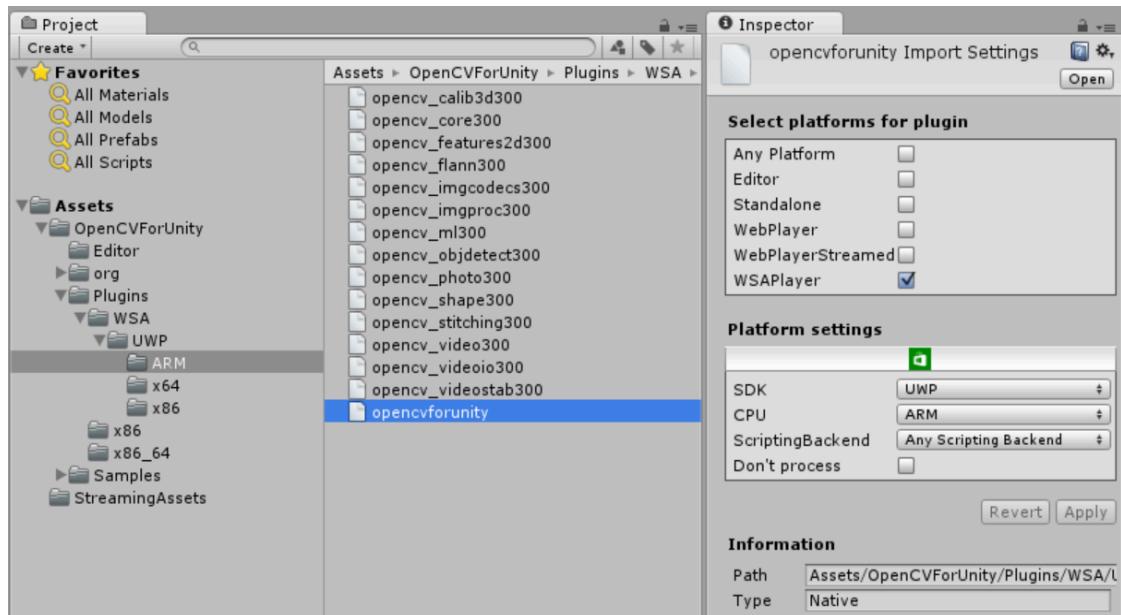
Mac Standalone Setup Procedure :

- “OpenCVForUnity/Plugins/macOS/opencvforunity.bundle” – Select platform Editor,Standalone and CPU x86_64 and OS OSX in Inspector.

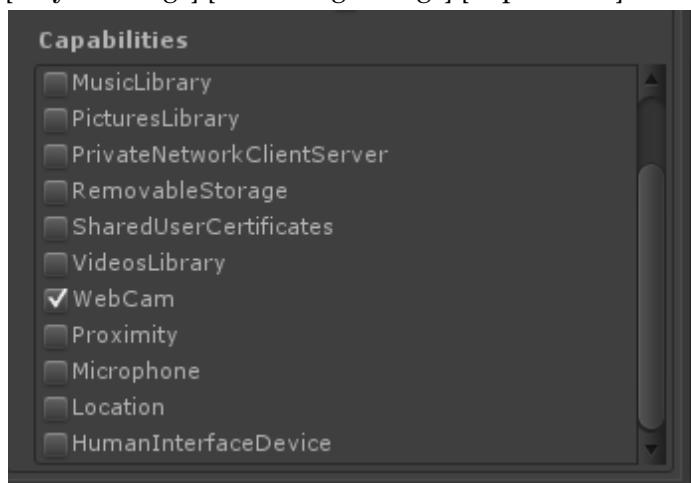


UWP Setup Procedure :

- “OpenCVForUnity/Plugins/WSA/UWP/ARM/*.dll” – Select platform WSAPlayer and SDK81 and CPU ARM in Inspector. Set “x86” and “x64” in the same way as “ARM”.

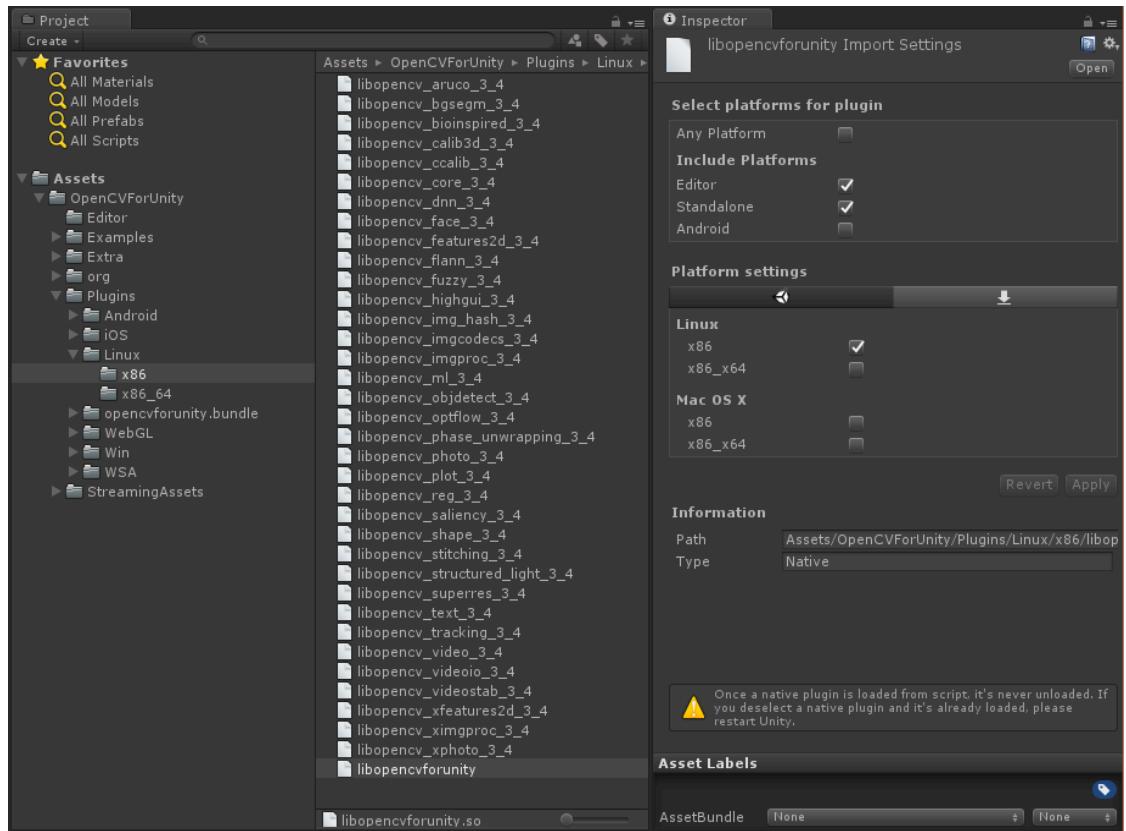


- If use webCamTexture class, Please choose “WebCam” in [PlayerSettings]-[PublishingSettings]-[Capabilities].

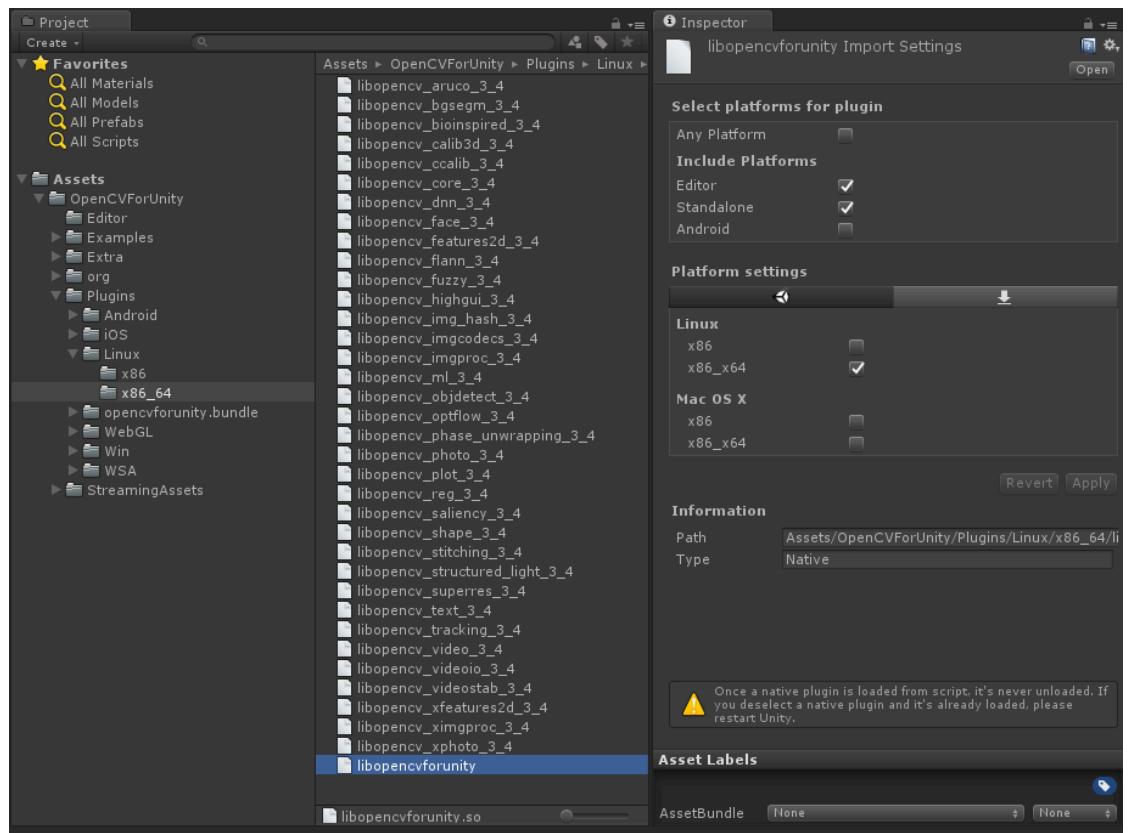


Linux Setup Procedure :

- “OpenCVForUnity/Plugins/Linux/x86/libopencvforunity.so” – Select platform Editor,Standalone and CPU x86 and OS Linux in Inspector.

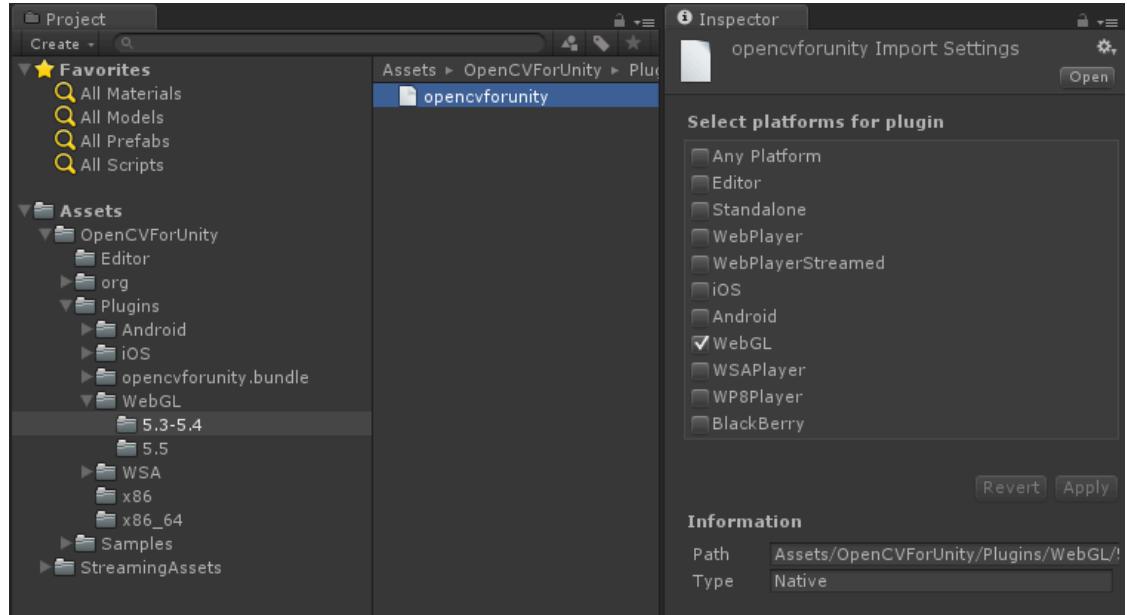


- “OpenCVForUnity/Plugins/Linux/x86_64/libopencvforunity.so” – Select platform Editor,Standalone and CPU x86_64 and OS Linux in Inspector.



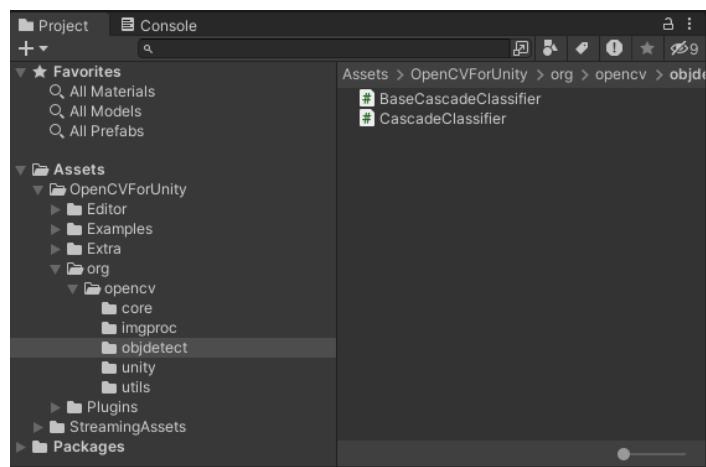
WebGL Setup Procedure :

- “OpenCVForUnity/Plugins/WebGL/unity_version/opencvforunity.bc” – Select platform WebGL in Inspector. You can easily set up the ImportSettings by clicking the [Set Plugin importSettings] button in the SetupTools window.



- Put the file that you want to use for Utils.getFilePathAsync() in the “Assets/StreamingAssets/”. In Case of WebGL platform, you need to use Utils.getFilePathAsync() instead of Utils.getFilePath(). (haarcascade_frontalface_alt.xml is for OpenCVForUnityExample.scene. Please copy only when necessary.)
- In the WebGL (asm.js) platform, the calculation result of Float type may be significantly different from other platforms. When using the OpenCV's method that use the Mat class (CvType is CV_32F) as an argument, you need to pay attention to the calculation precision.
- If you want to reduce the build size of your app to the bare minimum, try the following steps.
 1. PlayerSettings setting items related to build size optimization. “Code Optimization : Size”, “ il2cppCodeGeneration : Faster (smaller) builds”, “managedStrippingLevel : High”, “DecompreetionFormat : Brotli”
 2. Delete unused code from the "Assets/OpenCVForUnity/org" folder.

FaceDetectionExample leaving only the files necessary for it to work, and the result is as follows. Code Optimization : Size il2cppCodeGeneration : Faster (smaller) builds managedStrippingLevel : High DecompreetionFormat : Brotli result 6.31 MB (6,622,861 bytes) "Build" folder only 5.41 MB (5,681,838 bytes)



How to use OpenCV Dynamic Link Library with customized build settings :

- Download OpenCV 4.10.0 repository (git: [opencv](#), [opencv-contrib](#)).

Android

1. Change this line from "c++_shared" to "c++_static".
https://github.com/opencv/opencv/blob/4.x/platforms/android/build_sdk.py#L134
2. Build the Android SDK with “opencv/platforms/android/build_sdk.py”. (APP_STL := c++_static)

```
python ..\opencv\platforms\android\build_sdk.py ..\build ..\opencv  
--ndk_path=C://android-ndk --sdk_path=C://android-sdk  
--extra_modules_path=..\opencv_contrib\modules  
--config=ndk-18-api-level-24.config.py --no_samples_build --no_kotlin
```

3. Copy the output file (native\libs\arm64-v8a\libopencv_java4.so) to “OpenCVForUnity\Plugins\Android\libs\arm64-v8a\”. Copy the output files (native\libs\armeabi-v7a\libopencv_java4.so) to “OpenCVForUnity\Plugins\Android\libs\armeabi-v7a\”. Copy the output files (native\libs\x86\libopencv_java4.so) to “OpenCVForUnity\Plugins\Android\libs\x86\”. Copy the output files (native\libs\x86_64\libopencv_java4.so) to “OpenCVForUnity\Plugins\Android\libs\x86_64\”.
4. Copy “OpenCVForUnity\Extra\dll_version\Android\libs” to “OpenCVForUnity\Plugins\Android\libs\”.

iOS

1. Build the iOS framework with “opencv/platforms/ios/build_framework.py”.

```
python opencv/platforms/ios/build_framework.py --contrib opencv_contrib  
--dynamic ios
```

2. Copy the output file (opencv2.framework) to “OpenCVForUnity\Plugins\iOS\”.

Windows

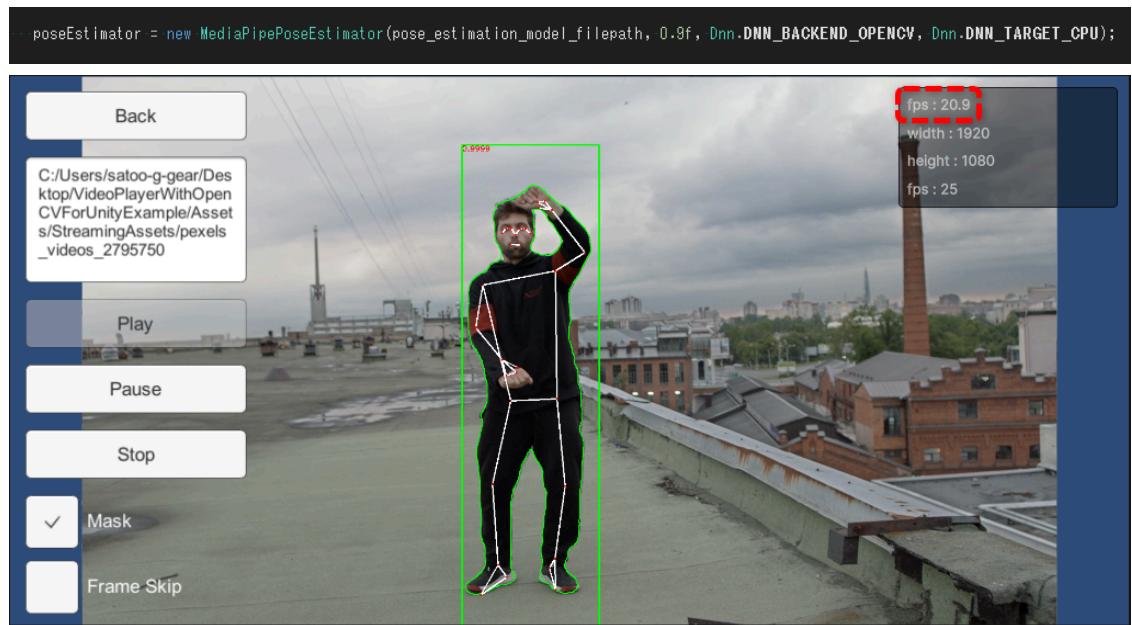
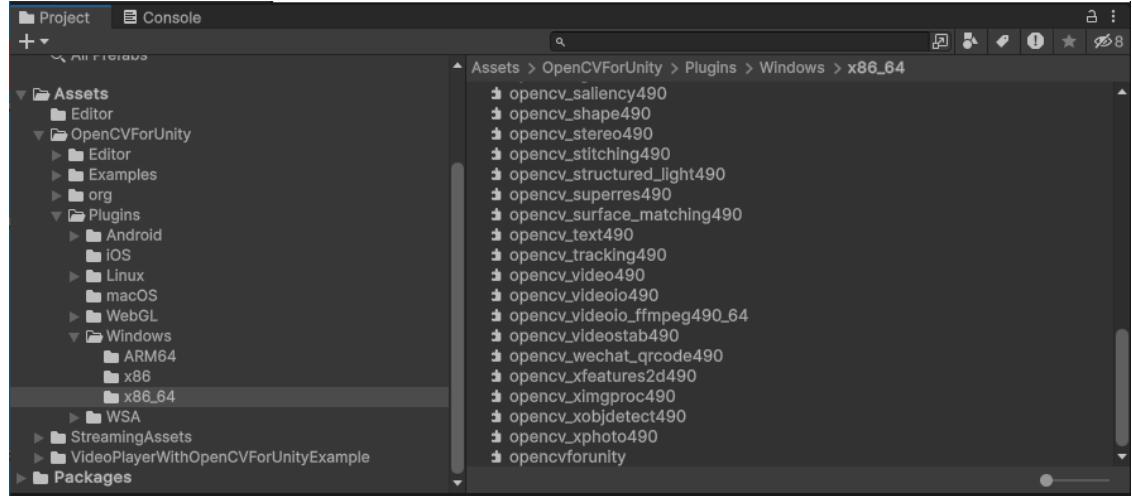
1. Build the OpenCV dynamic library.

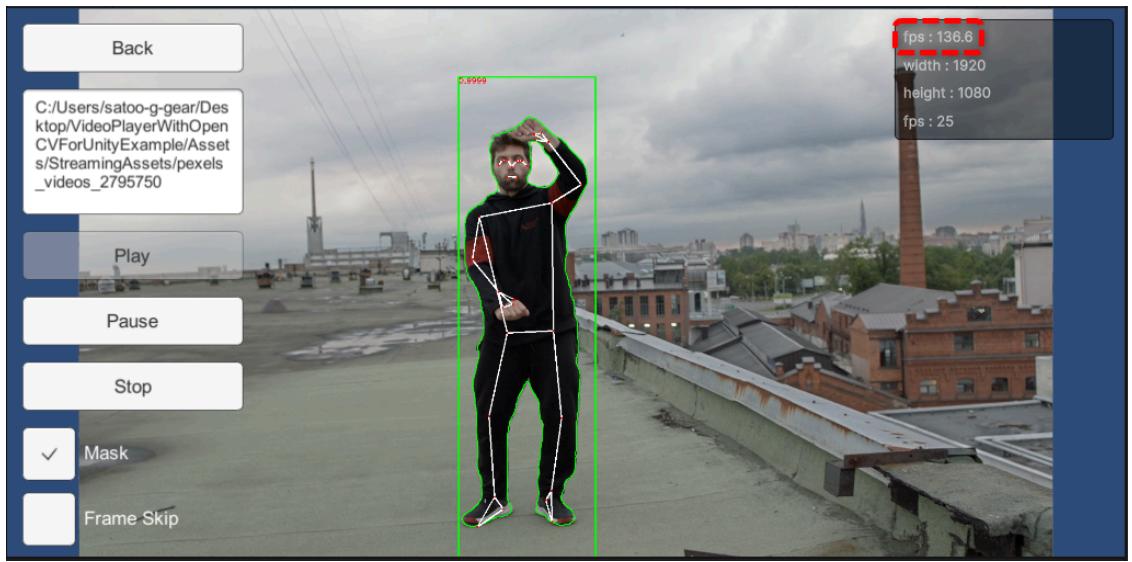
```
OPENCV_EXTRA_MODULES_PATH:PATH=C:/Users/xxxxx/opencv_contrib/mod  
ules  
BUILD_SHARED_LIBS:BOOL=ON
```

2. Copy “install\x64\vc16\bin\” to “OpenCVForUnity\Plugins\Windows\x64\”. Copy “OpenCVForUnity\Extra\dll_version\Windows\” to “OpenCVForUnity\Plugins\Windows\”.

Comparison of inference speed on Windows platform using the dnn module enabled the CUDA backend.

<https://enoxsoftware.com/opencvforunity/how-to-run-dnn-modules-with-cuda-backend-support-on-windows-platform/>





macOS

1. Build the OpenCV library.

```
CMAKE_OSX_ARCHITECTURESSTRING=x86_64
OPENCV_EXTRA_MODULES_PATHPATH=/Users/xxxxxx/opencv_contrib/modules
```

2. Copy the output files (libopencv_*.4.10.0.dylib) to opencvforunity.bundle\Contents\MacOS\.
3. Relink libopencv_*.4.10.0.dylib using otool and install_name_tool.
<http://phenixyu.blogspot.com/2016/09/how-to-load-dynamic-library-with-unity.html>

```
//example : aruco module
otool -L libopencv_aruco.4.10.0.dylib

install_name_tool -id @loader_path/libopencv_aruco.4.10.0.dylib
libopencv_aruco.4.10.0.dylib

install_name_tool -change @rpath/libopencv_calib3d.4.10.dylib
@loader_path/libopencv_calib3d.4.10.0.dylib libopencv_aruco.4.10.0.dylib

install_name_tool -change @rpath/libopencv_features2d.4.10.dylib
@loader_path/libopencv_features2d.4.10.0.dylib libopencv_aruco.4.10.0.dylib

install_name_tool -change @rpath/libopencv_flann.4.10.dylib
@loader_path/libopencv_flann.4.10.0.dylib libopencv_aruco.4.10.0.dylib
```

```
install_name_tool -change @rpath/libopencv_highgui.4.10.dylib  
@loader_path/libopencv_highgui.4.10.0.dylib libopencv_aruco.4.10.0.dylib

install_name_tool -change @rpath/libopencv_videoio.4.10.dylib  
@loader_path/libopencv_videoio.4.10.0.dylib libopencv_aruco.4.10.0.dylib

install_name_tool -change @rpath/libopencv_imgcodecs.4.10.dylib  
@loader_path/libopencv_imgcodecs.4.10.0.dylib libopencv_aruco.4.10.0.dylib

install_name_tool -change @rpath/libopencv_imgproc.4.10.dylib  
@loader_path/libopencv_imgproc.4.10.0.dylib libopencv_aruco.4.10.0.dylib

install_name_tool -change @rpath/libopencv_core.4.10.dylib  
@loader_path/libopencv_core.4.10.0.dylib libopencv_aruco.4.10.0.dylib
```

Linux

1. Build the OpenCV library.
2. Rename output files (libopencv_*.so.4.10.0).

```
sudo apt-get install rpl  
sed -i -e 's/.so\410/_410.so/g' ./libopencv_*.so.4.10.0  
rename "s/.so.4.10.0/_410.so/" libopencv_*.so.4.10.0
```

3. Copy libopencv_*_4_10.so to “/OpenCVForUnity/Plugins/Linux/x86_64” folder.

UWP

1. Build the OpenCV library.

```
cd C:\Users\satoo\Desktop\opencv\platforms\winrt  
setup_winrt.bat "WS" "10.0" "x64"
```

2. Copy “install\WS\10.0\ARM\ARM\vc16\bin” to “OpenCVForUnity\Plugins\WSA\UWP\ARM”. Copy “install\WS\10.0\x64\x64\vc16\bin” to “OpenCVForUnity\Plugins\WSA\UWP\x64”. Copy “install\WS\10.0\x86\x86\vc16\bin” to “OpenCVForUnity\Plugins\WSA\UWP\x86”.

Q & A

Q1.

Asset package size is large. Is there a way to reduce?

A1.

Please remove plugin folders of non-output target platforms that are included in OpenCVforUnity package. You do not need to import plug-in files for platforms not supported by your project.

Q2.

Support Web platform?

A2.

Since the Unity Web Player does not support the native plugin, "OpenCV for Unity" does not support "WebPlayer Platform".

A WebGL platform was added as an alternative. (Unity 5.3 or higher).

Q3.

How do learn the details of OpenCV's method and argument?

A3.

Please refer to OpenCV official document (<http://docs.opencv.org/4.5.5/index.html>) and OpenCV Tutorials (http://docs.opencv.org/4.5.5/d9/df8/tutorial_root.html) for the details of the argument of the method..

Q4.

How can I convert Mat class operators defined in C++?

A4.

Way to translation of Mat class operators defined in C++.

<https://enoxsoftware.com/opencvforunity/way-to-translation-of-mat-class-operators-defined-in-cpp/>

Q5.

"DllNotFoundException: opencvforunity" is displayed on the console when run the example scene.

A5.

The plugin does not seem to be loaded correctly. Please check the setup procedure.

Q6.

"ArgumentException: The output Mat object has to be of the same size" is displayed on the console when run the example scene.

A6.

After having setup Plugin, Plugin may work well when you reboot Unity.

Q7.

"Level 'Texture2DtoMatExample' (-1) could not be loaded because it has not been added to the build settings." is displayed on the console when run the example scene.

A7.

Please add all of "***.unity" scenes into the "Assets/OpenCVForUnity/Examples" folder to

[Build Settings] – [Scene In Build].

Q8.

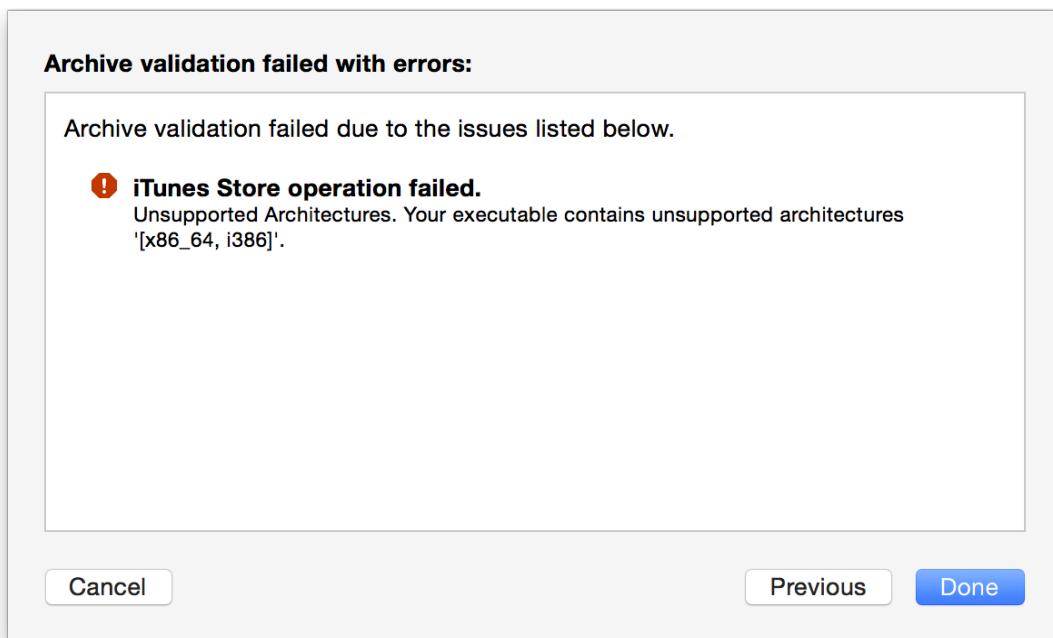
In DetectFaceExample or WebCamTextureDetectFaceExample, red rectangle is not displayed around a face.

A8.

You might have failed to read the “haarcascade_frontalface_alt.xml”. Please confirm whether there is the “StreamingAssets” folder at the right position.

Q9.

[iOS]Submit to App Store issues: Unsupported Architecture x86, i386“Unsupported Architecture. Your executable contains unsupported architecture ‘[x86_64, i386]’.”



A9.

<http://ioscake.com/submit-to-app-store-issues-unsupported-architecture-x86.html>

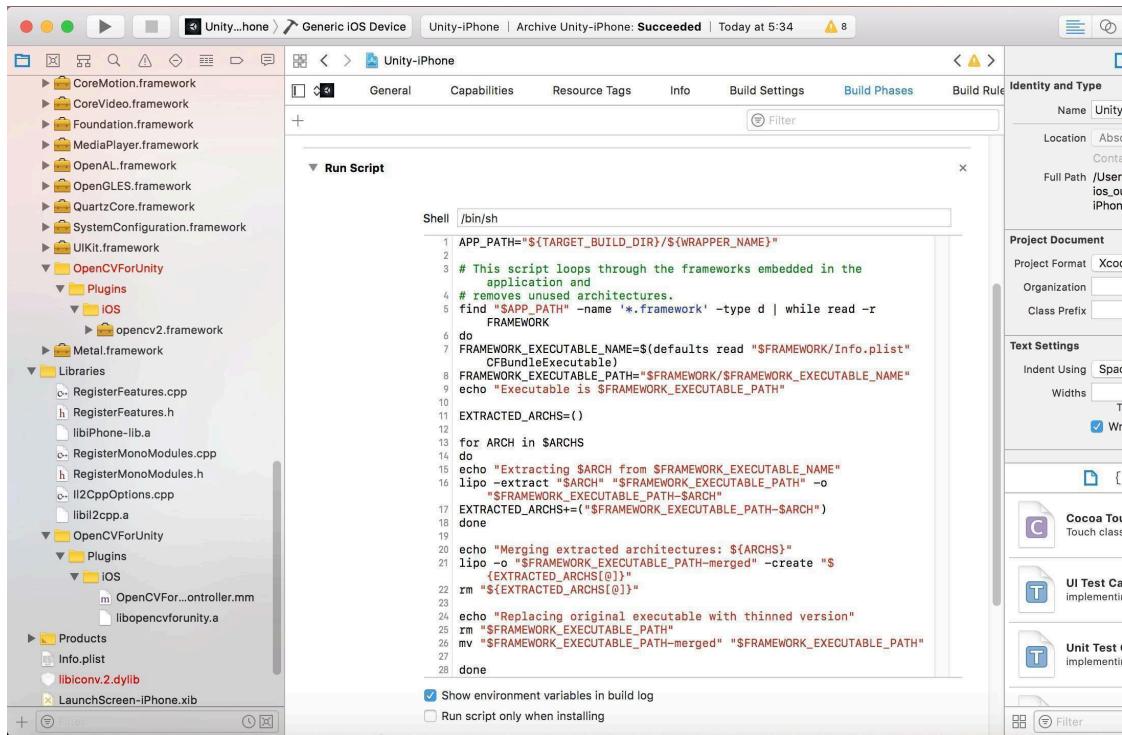
“The problem is that the Buy framework contains a build for both the simulator (x86_64) and the actual devices (ARM).

Of course, you aren’t allowed to submit to the App Store a binary for an unsupported architecture, so the solution is to “manually” remove the unneeded architectures from the final binary, before submitting it.”

There are **two ways** to solve this error.

1. Please add the script of this page to BuildPhases->RunScript.

<http://ikennd.ac/blog/2015/02/stripping-unwanted-architectures-from-dynamic-libraries-in-xcode/>



2. Please execute the following command on terminal.

<https://stackoverflow.com/questions/42641806/check-and-remove-unsupported-architecture-x86-64-i386-in-ipa-archive>

```

//remove i386 architectures.
lipo -remove i386 opencv2.framework/opencv2 -o opencv2.framework/opencv2

//remove x86_64 architectures.
lipo -remove x86_64 opencv2.framework/opencv2 -o opencv2.framework/opencv2

//check the architectures.
lipo -info opencv2.framework/opencv2

```

Q10.

How can I use SIFT or SURF algorithms?

A10.

The native library included in OpenCVForUnity is built with the OPENCV_ENABLE_NONFREE flag disabled. To use the SIFT and SURF algorithms, rebuild OPENCV library with OPENCV_ENABLE_NONFREE enabled. For more details, see the section on “How to use OpenCV Dynamic Link Library with customized build settings” in this document.

Q11.

How to catch native OpenCV's errors code (CVException handling)

A11.

In order to display the native opencv's error code, please enclose the code in `Utils.setDebugMode(true)` and `Utils.setDebugMode(false)`.

```
Utils.setDebugMode(true); ----- Utils.setDebugMode(false);
```

See this page for details. ([\[Tips\] How to catch native OpenCV's errors code \(CVException handling\)](#))

Q12.

What is the minimum file composition required for the assets to work?

A12.

You do not necessarily have to import all the files for the asset to work.

If you do not need to try the example scenes, the minimum file composition required is as follows:

