

OpenCV for Unity 3.0.0



About OpenCV for Unity

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

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

Our Asset Features

Cross Platform

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

- **Mobile:** iOS & Android
- **Desktop:** Windows & macOS & Linux Standalone support
- **Web:** WebGL support
- **ChromeOS:** ChromeOS support
- **XR:** visionOS support (beta) & Windows10 UWP support
- **Editor:** Support for preview in the Editor

Support Latest Version

- Users can utilize **OpenCV for Unity** based on the latest version of **OpenCV (4.12.0)** for app development. (git: [opencv](#), [opencv-contrib](#))

OpenCV Java API Support

- As a **clone of OpenCV Java**, which means you can use the exact **same API as OpenCV Java 4.12.0**
- **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 example 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 \(GitHub\)](#)

Example Code using OpenCV for Unity

AR & Recognition Examples

- [MarkerBased AR Example](#)
- [MarkerLess AR Example](#)
- [FaceMask Example](#)
- [RealTime FaceRecognition Example](#)

Integration Examples

- [AVPro with OpenCV for Unity Example](#)
- [Kinect with OpenCV for Unity Example](#)
- [HoloLens with OpenCV for Unity Example](#)
- [VisualScriptingWithOpenCVForUnityExample](#)
- [ARFoundationWithOpenCVForUnityExample](#)
- [NrealLight with OpenCVForUnity Example](#)

Media & AI Examples

- [FfmpegWithOpenCVForUnityExample](#)
 - [VideoPlayerWithOpenCVForUnityExample](#)
 - [NativeGalleryWithOpenCVForUnityExample](#)
 - [YOLOv5WithOpenCVForUnityExample](#)
 - [YOLOv8WithOpenCVForUnityExample](#)
 - [KlakNDIWithOpenCVForUnityExample](#)
-

Quick Links

- **Official Site** - Main product page
 - **Example Code** - GitHub repository with examples
 - **Android Demo** - Download Android demo app
 - **WebGL Demo** - Try WebGL demo online
 - **Product Introduction Video** - Watch product overview
 - **Setup Guide Video** - Step-by-step setup tutorial
 - **Forum** - Community discussions
 - **API Reference** - Complete API documentation
 - **Support Modules** - Additional module documentation
-

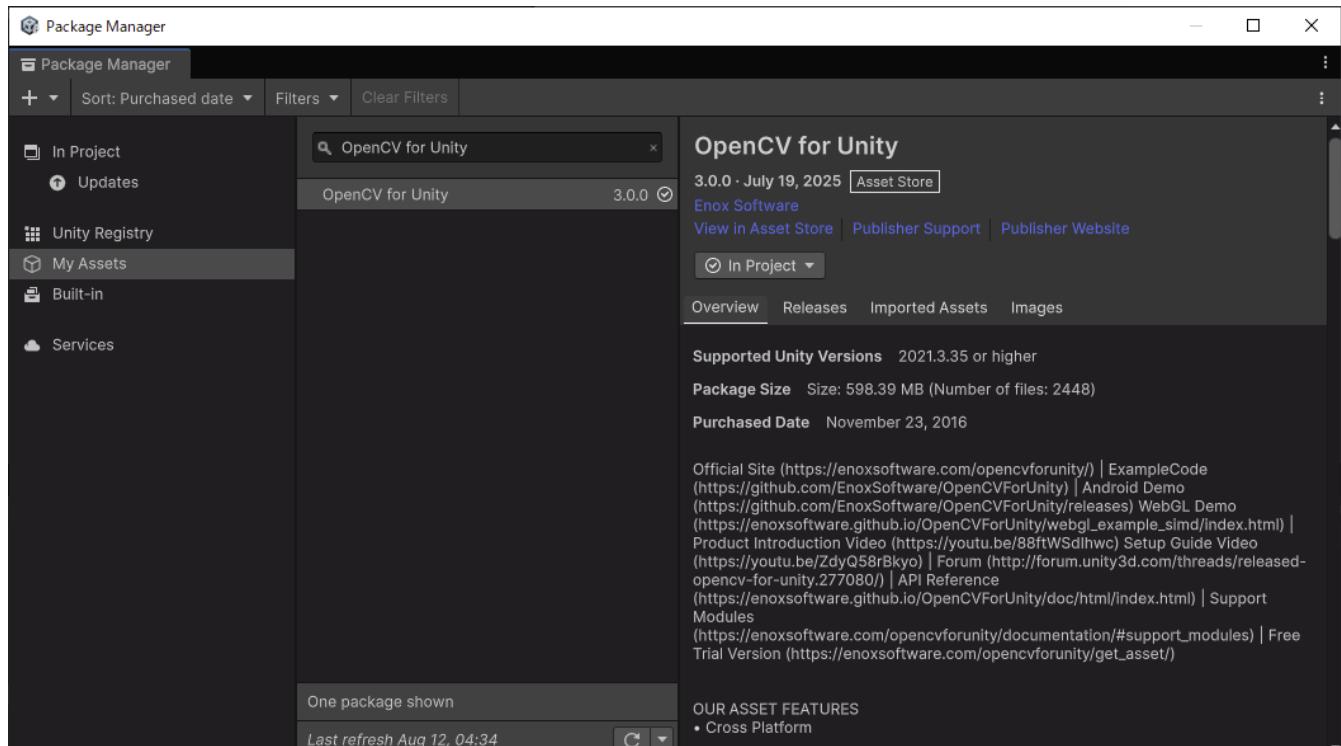
System Requirements

Platform	Requirement
Windows Standalone & Editor	Windows 8 or later
Mac Standalone & Editor	macOS 10.13 or later
Linux Standalone & Editor	Ubuntu 18.04 or later
Android	API level 24 or later
iOS	iOS Version 12.0 or later
visionOS (beta)	visionOS 1 or later

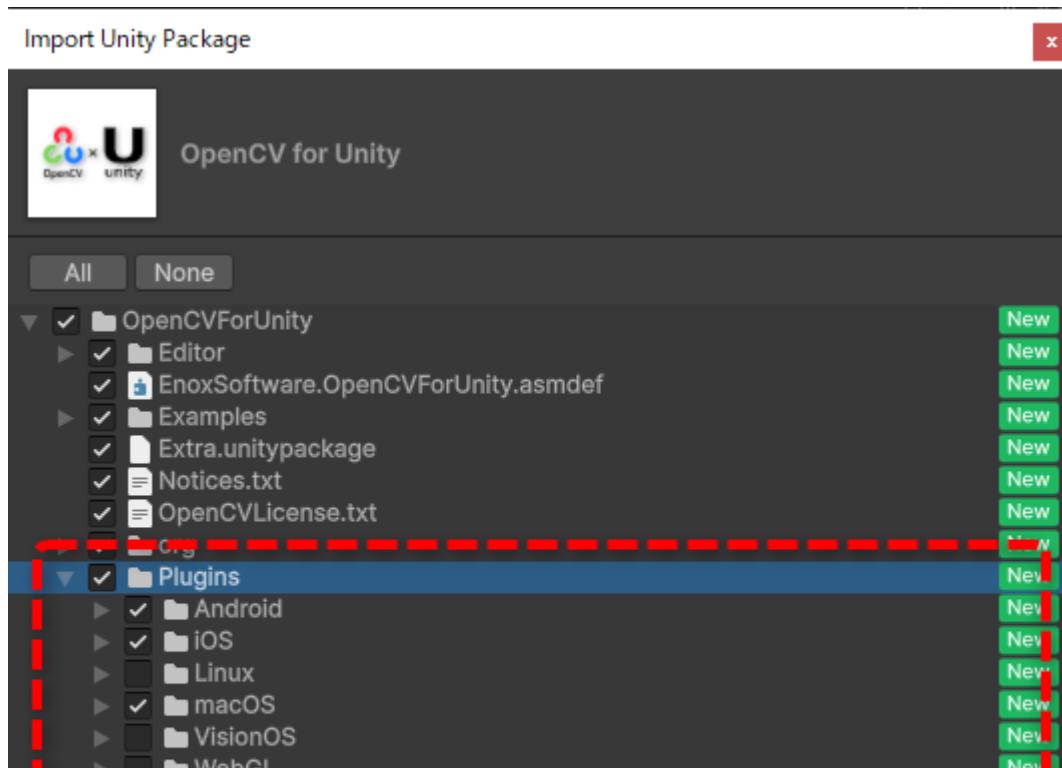
Setup Guide for Getting Started ([Setup Guide Video](#))

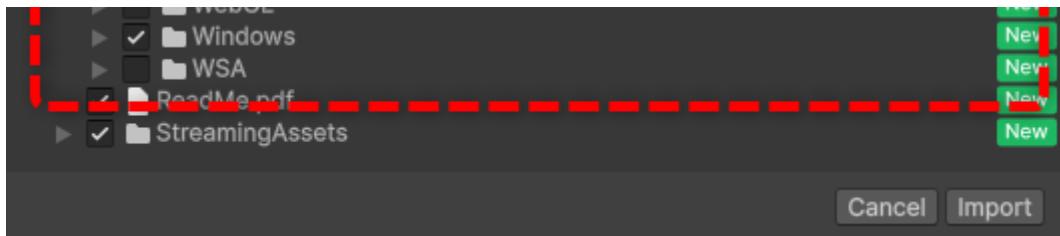
1. Import **OpenCV for Unity** from the **Package Manager**.

⚠ Important: If your project already has a previous version of the `Assets/OpenCVForUnity` folder, issues may occur. Please delete the `Assets/OpenCVForUnity` folder first before importing the new version.



At that time, by unchecking platforms other than those planned to be supported in your development project, unnecessary plugin files for those platforms will not be imported, reducing the project size.

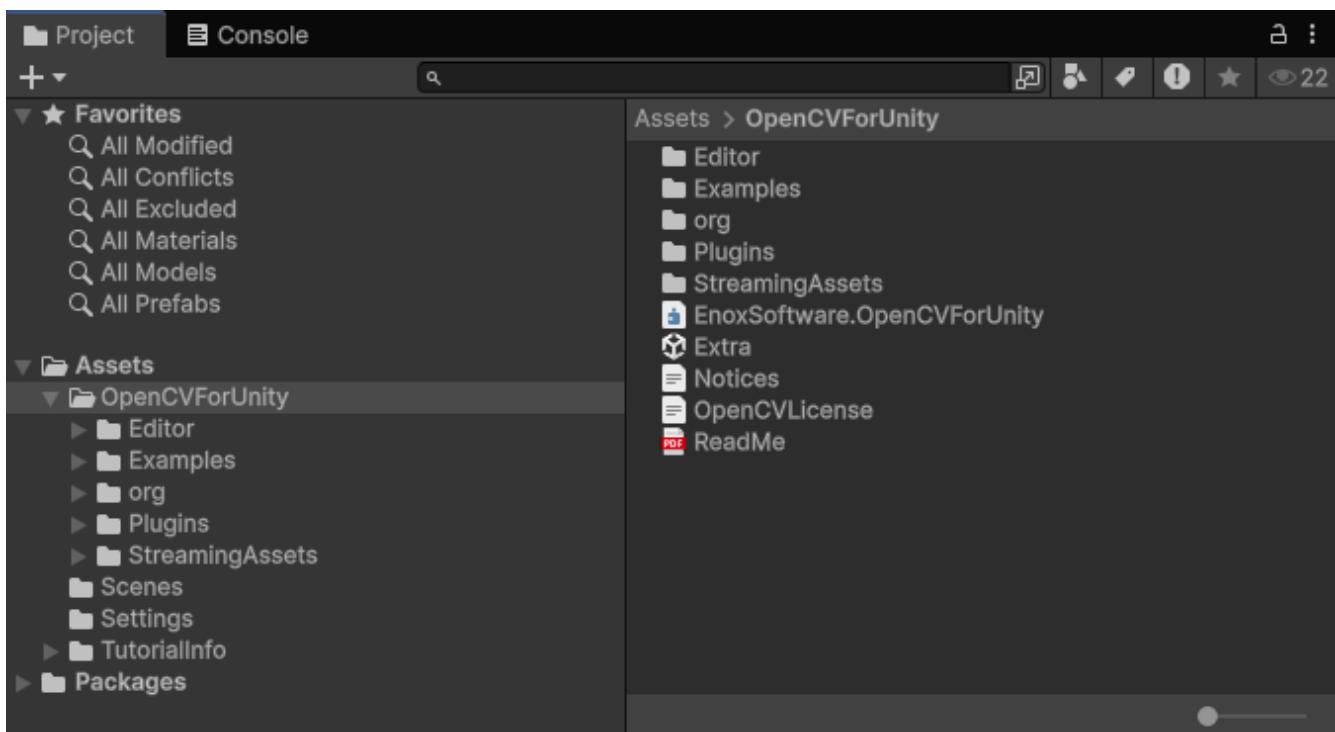




Even after import, you can easily reduce unnecessary files in the `Assets/OpenCVForUnity` folder using the `Project Size Reducer`. Particularly, by removing plugin files for platforms not planned to be supported in your development project, you can significantly reduce the project size. For details, please see [Project Size Reducer Window Description](#).

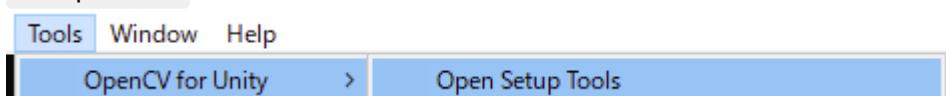
2. When the import is complete, the `Setup Tools` window will automatically appear. In the `Setup Tools` window, you can easily perform various settings. For details, please see [Setup Tools Window Description](#).
3. Setup is now complete.

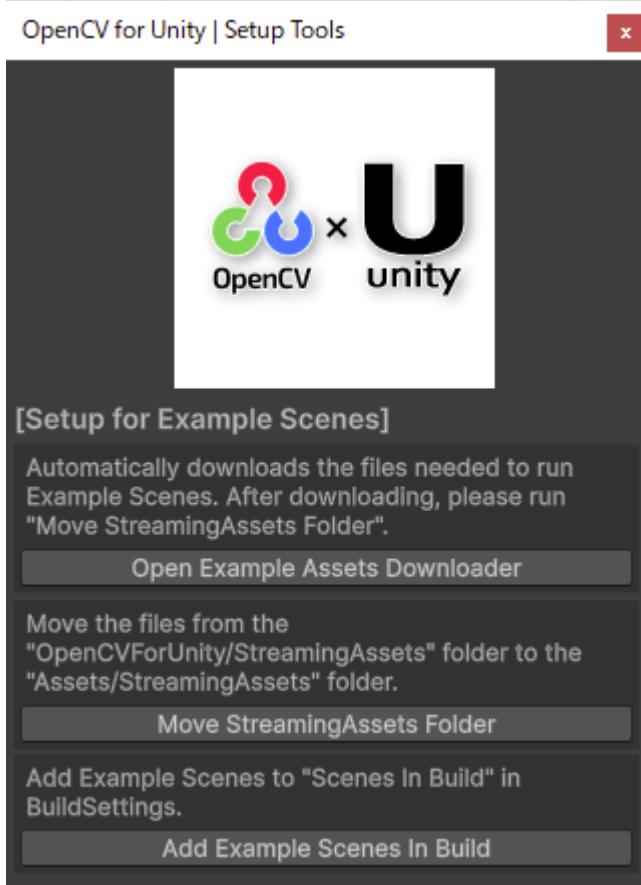
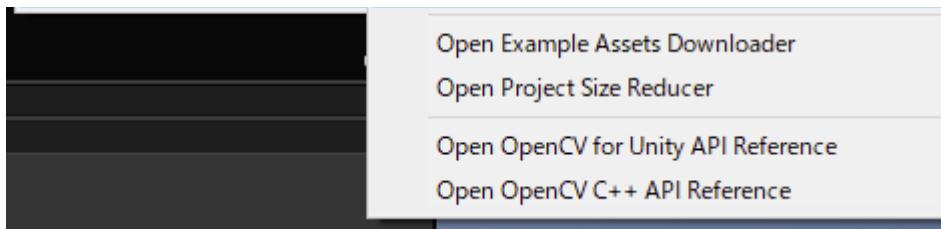
As a starting point for development using this asset, we recommend first referring to the API usage examples in the Example Scenes. Also, since the OpenCV for Unity API is the same as the OpenCV Java API, OpenCV Java tutorials are also helpful.



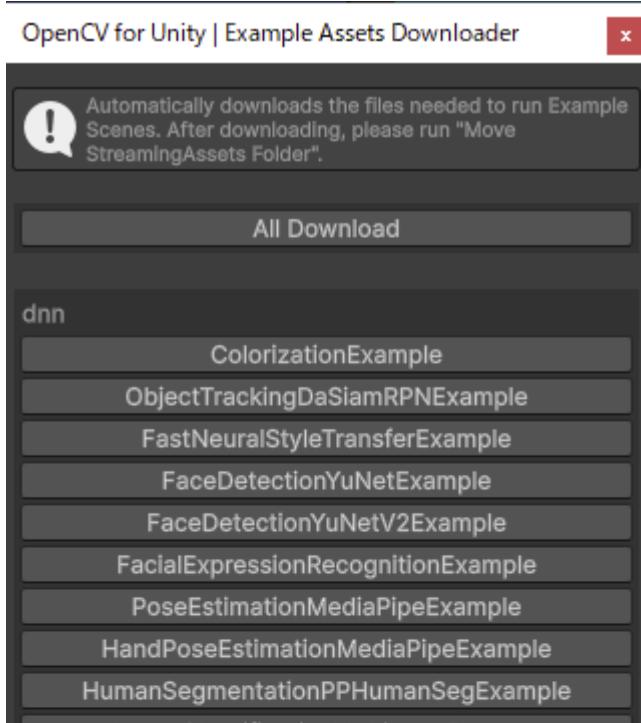
Example Scenes Setup Guide ([Setup Guide Video](#))

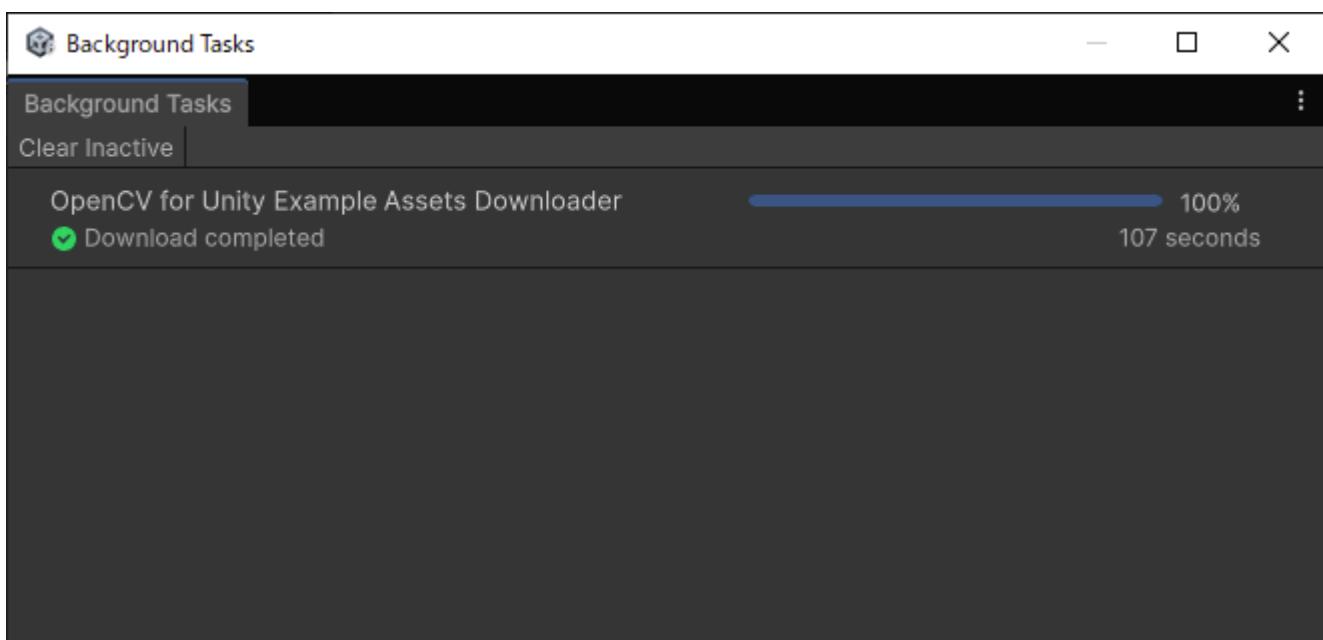
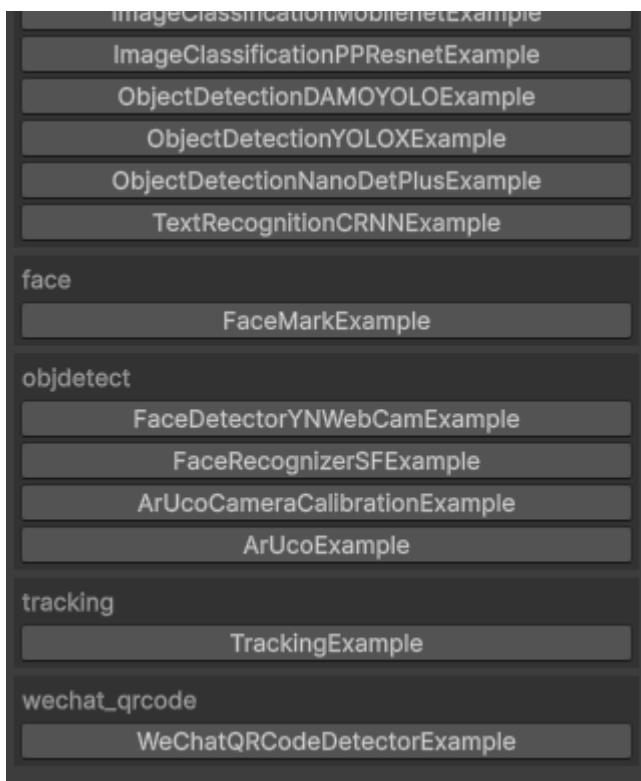
1. Complete the [Setup Guide for Getting Started](#).
2. Select the menu item `Tools > OpenCV for Unity > Open Setup Tools` to open the `Setup Tools` window.



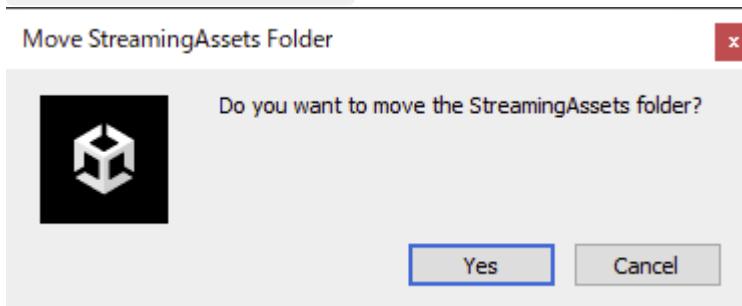


3. Click the `Open Example Assets Downloader` button in the `Setup Tools` window to open the `Example Assets Downloader` window, then click the `All Download` button to download the necessary files for Example Scenes. [Example Assets Downloader Window Description](#)

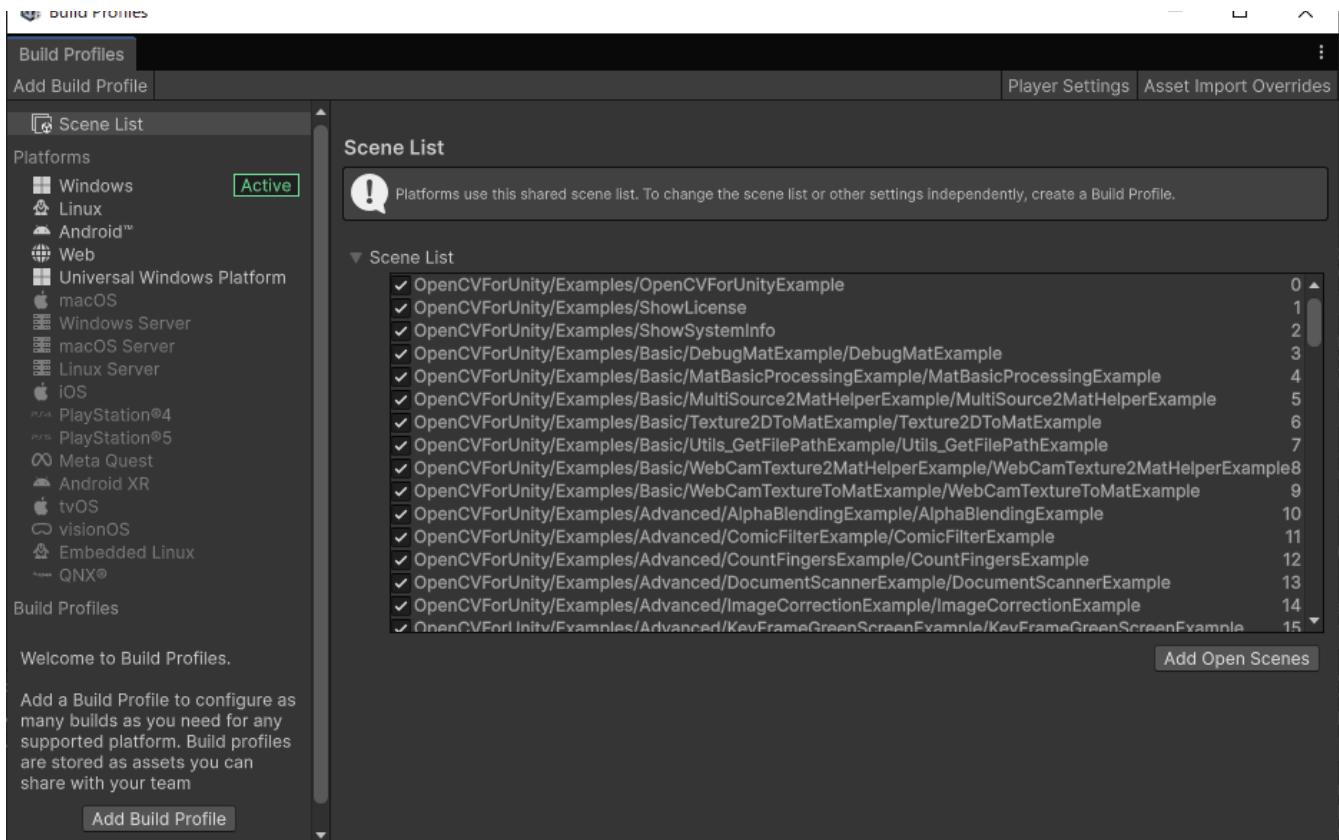




4. Click the `Move StreamingAssets Folder` button in the `Setup Tools` window to move the `Assets/OpenCVForUnity/StreamingAssets/OpenCVForUnityExamples` folder to directly under the `Assets/StreamingAssets` folder.



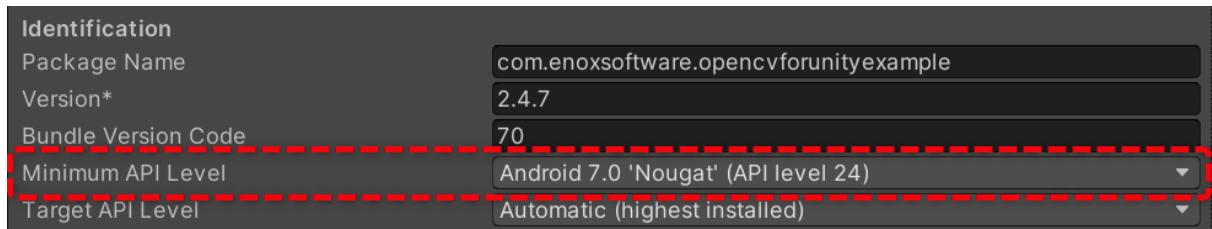
5. Click the `Add Example Scenes in Build` button in the `Setup Tools` window to add all Example Scene files in the `Examples` folder to `Scenes In Build` in `Build Settings`.



6. When building Example Scenes and running them on devices, settings may need to be changed depending on the platform.

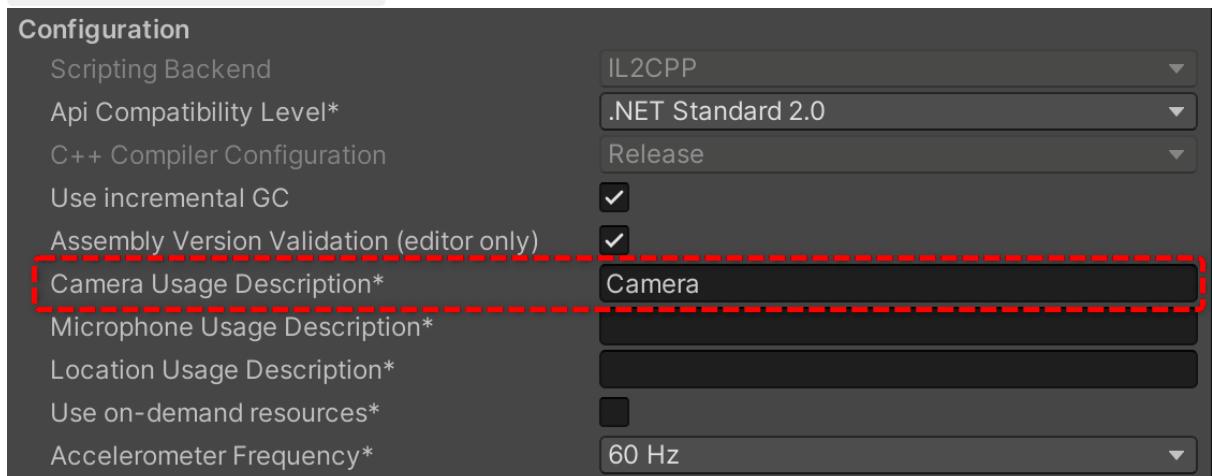
- **Android**

- Set **Player Settings** → **Other Settings** → **Identification** → **Minimum API Level** to **24** or higher.



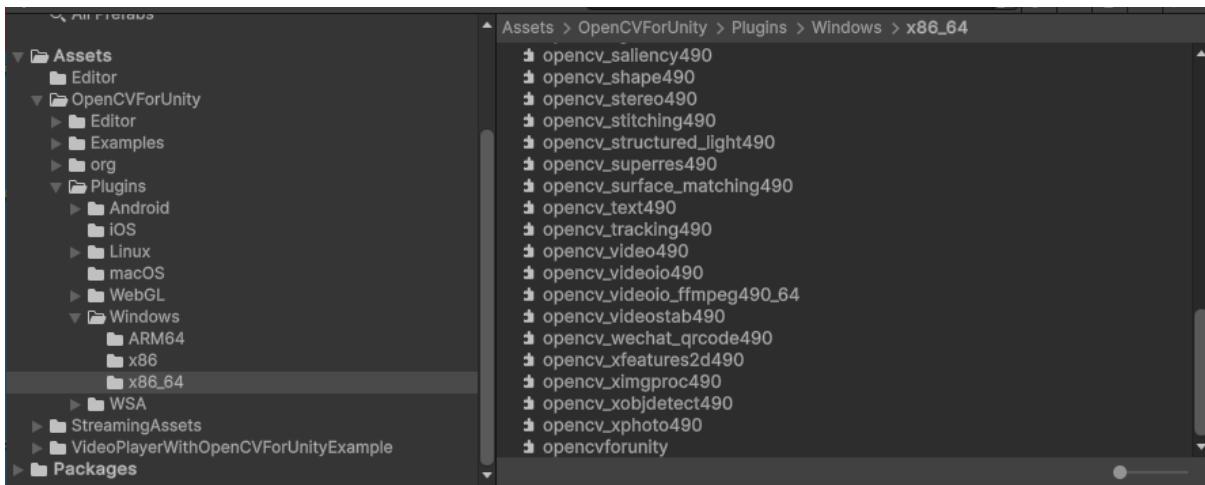
- **iOS**

- Set **Player Settings** → **Other Settings** → **Configuration** → **Camera Usage Description**.



- Set **Target minimum iOS Version** to **12.0** or higher.

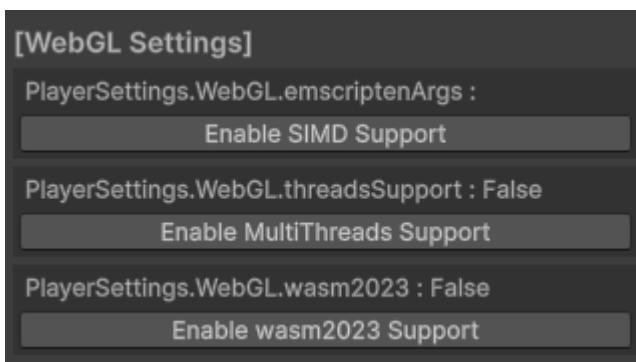




- **WebGL**

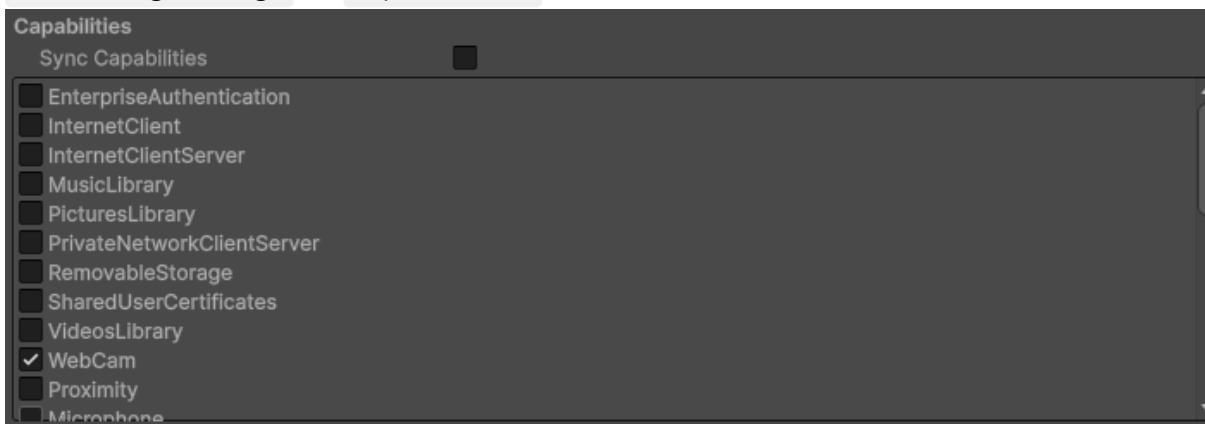
- If you want to enable multi-threading or SIMD optimization, click the `Enable MultiThreads Support` button or `Enable SIMD Support` button in the `Setup Tools` window. [Setup Tools Window Description](#)

⚠ The browser you run must support multi-threading and SIMD.

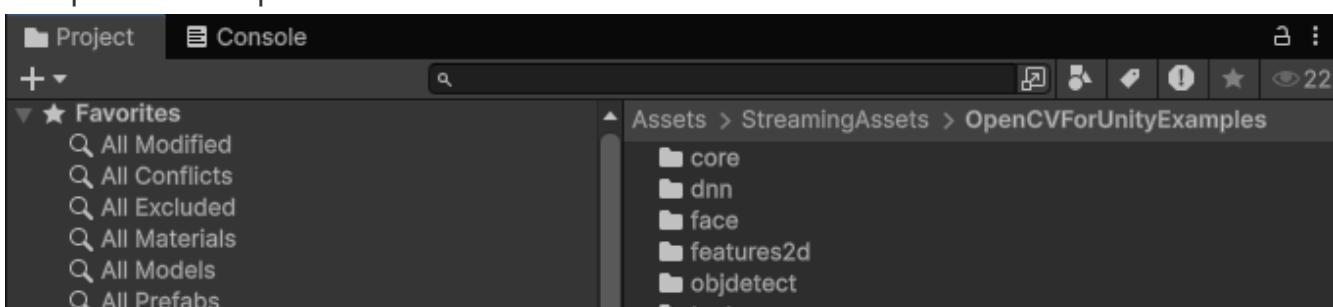


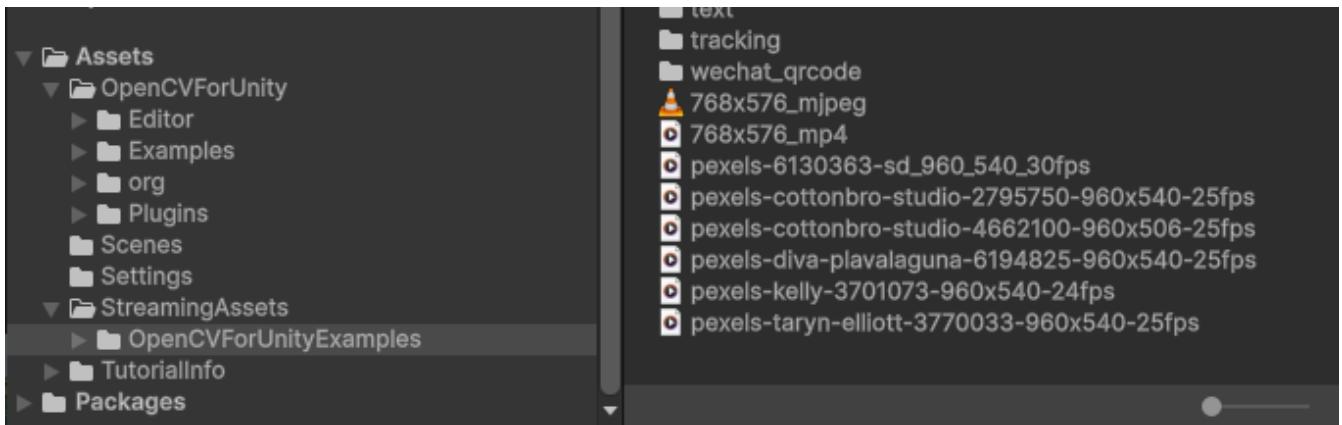
- **Windows10 UWP**

- If using the `WebCamTexture` class, check `WebCam` in `Player Settings` → `PublishingSettings` → `Capabilities`.



7. Setup is now complete.





For each Example, please see [Example Scenes Description](#).

Example Scenes Description

The Example Scenes in **OpenCV for Unity** are structured to allow you to learn various **OpenCV** features progressively. They are divided into the following **4** categories, and by starting from **Basic** and progressing to **Advanced**, you can learn **OpenCV** features practically.

Basic Folder

Contains Examples related to **basic functionality and setup**:

- **Texture2DToMatExample**: Example of converting Unity's `Texture2D` class to OpenCV's `Mat` class
- **WebCamTextureToMatExample**: Basic example of converting Unity's `WebCamTexture` class video to `Mat` class
- **WebCamTexture2MatHelperExample**: Example of using a helper class to more efficiently convert Unity's `WebCamTexture` class video to `Mat` class
- **MultiSource2MatHelperExample**: Example of using a helper class to switch between multiple video acquisition sources in the `inspector` and convert them to `Mat` class
- **MatBasicProcessingExample**: Example showing basic processing methods of `Mat` class
- **DebugMatExample**: Example of `Mat` class debug display functionality
- **Utils_GetFilePathExample**: Example of using file path acquisition utility

MainModules Folder

Contains Examples related to **OpenCV's main modules**:

Important Modules

- `core` : Basic functionality

- **PCAExample**: Principal component analysis implementation example
- **KMeansClusteringExample**: K-means clustering implementation example
- `imgproc` : Image processing
 - **ThresholdExample**: Threshold processing example
 - **DrawingExample**: Shape drawing functionality example
 - **HoughLinesPExample**: Line detection example
 - **CircleDetectionExample**: Circle detection example
 - **ConvexHullExample**: Convex hull calculation example
 - **ConnectedComponentsExample**: Connected component analysis example
 - **MatchTemplateExample**: Template matching example
 - **MatchShapesExample**: Shape matching example
 - **GrabCutExample**: Foreground extraction example
 - **WrapPerspectiveExample**: Perspective transformation example
- `features2d` : Feature point detection
 - **FeatureMatchingExample**: Feature point matching example
 - **SimpleBlobExample**: Blob detection example
 - **MSERExample**: MSER (Maximally Stable Extremal Regions) detection example
 - **HomographyToFindAKnownObjectExample**: Known object detection using homography example
- `dnn` : Deep learning
 - **ObjectDetectionYOLOXExample**: Object detection using YOLOX example
 - **ObjectDetectionDAMOYOLOExample**: Object detection using DAMO-YOLO example
 - **ObjectDetectionNanoDetPlusExample**: Object detection using NanoDetPlus example
 - **ImageClassificationMobilenetExample**: Image classification using MobileNet example
 - **ImageClassificationPPResnetExample**: Image classification using PP-ResNet example
 - **FaceDetectionYuNetExample**: Face detection using YuNet example
 - **FaceDetectionYuNetV2Example**: Face detection using YuNetV2 example
 - **FacialExpressionRecognitionExample**: Facial expression recognition example
 - **HandPoseEstimationMediaPipeExample**: Hand pose estimation using MediaPipe example
 - **PoseEstimationMediaPipeExample**: Pose estimation using MediaPipe example
 - **HumanSegmentationPPHumanSegExample**: Human segmentation example
 - **TextRecognitionCRNNEExample**: Text recognition using CRNN example
 - **ObjectTrackingDaSiamRPNExample**: Object tracking using DaSiamRPN example
 - **FastNeuralStyleTransferExample**: Fast neural style transfer example
 - **ColorizationExample**: Automatic image colorization example

Other Modules

- `calib3d` : Camera calibration and 3D processing

- `imgcodecs` : Image loading and saving
- `video` : Video processing
- `videoio` : Video input/output
- `photo` : Image restoration and noise removal
- `objdetect` : Object detection
- `ml` : Machine learning

ContribModules Folder

Contains Examples related to **OpenCV's additional modules**:

- `face` : Face recognition and facial feature point detection
 - **FaceRecognizerExample**: Face recognition example
 - **FaceMarkExample**: Facial feature point detection example
- `wechat_qrcode` : QR code detection
 - **WeChatQRCodeDetectorExample**: WeChat QR code detector example
- `tracking` : Object tracking
- `text` : Text detection and recognition
- `plot` : Graph drawing
- `bgsegm` : Background segmentation

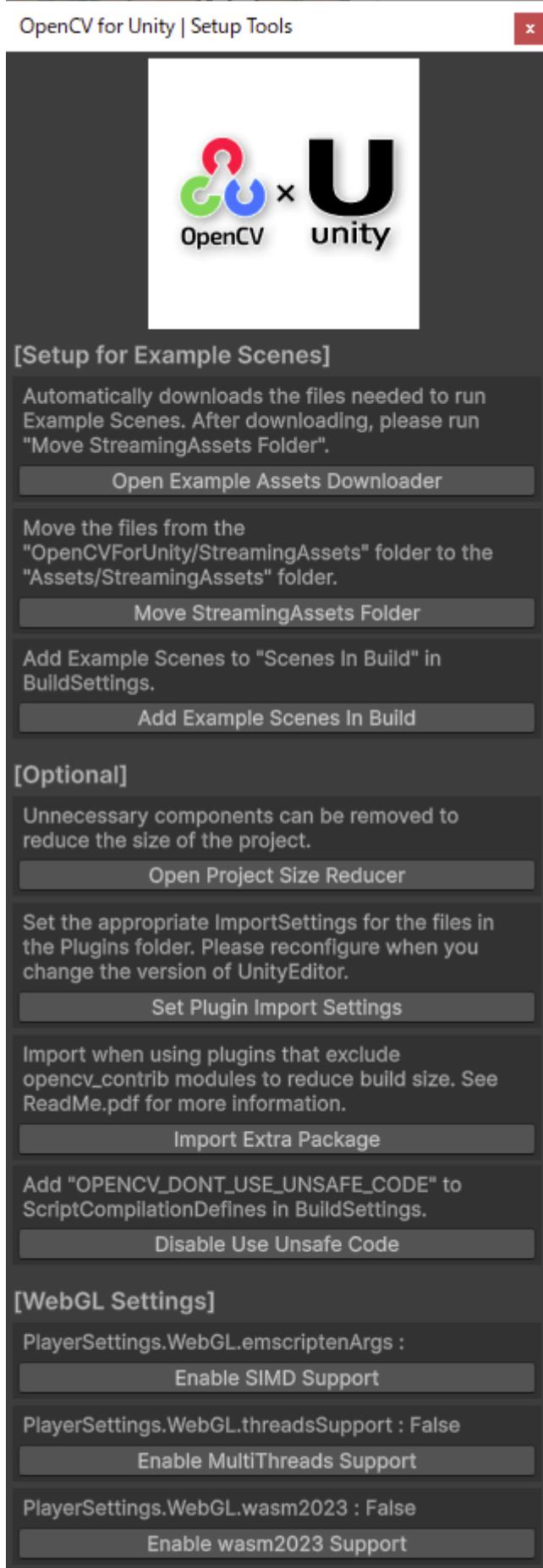
Advanced Folder

Contains **advanced implementation examples**:

- **AlphaBlendingExample**: Alpha blending implementation example
- **ComicFilterExample**: Comic-style filter implementation example
- **CountFingersExample**: Finger counting implementation example
- **DocumentScannerExample**: Document scanner implementation example
- **ImageCorrectionExample**: Image correction implementation example
- **KeyFrameGreenScreenExample**: Keyframe-based green screen implementation example
- **MultiObjectTrackingExample**: Multi-object tracking implementation example
- **MultiObjectTrackingBasedOnColorExample**: Color-based multi-object tracking implementation example
- **PhysicalGreenScreenExample**: Physics-based green screen implementation example
- **PolygonFilterExample**: Polygon filter implementation example

Setup Tools Window Description

Select the menu item `Tools > OpenCV for Unity > Open Setup Tools` to open the `Setup Tools` window.



This window is a collection of tools to make **OpenCV for Unity** project setup easier.

Each section has a specific role and is structured as follows.

Setup for Example Scenes

This section contains tools for preparing and setting up files necessary to run Example Scenes.

- **Open Example Assets Downloader**

Opens the `Example Assets Downloader` window that automatically downloads files necessary to run Example Scenes. For details, please see [Example Assets Downloader Window Description](#).

After downloading, please execute the next step `Move StreamingAssets Folder`.

- **Move StreamingAssets Folder**

Moves downloaded files from the `Assets/OpenCVForUnity/StreamingAssets` folder to the `Assets/StreamingAssets` folder.

- **Add Example Scenes In Build**

Automatically adds Example Scenes to `Scenes In Build` in `Build Settings`.

Optional

This section contains convenient tools for optional use, such as project size reduction and importing `Extra` packages.

- **Open Project Size Reducer**

Opens the `Project Size Reducer` window that removes unnecessary components and reduces project size. For details, please see [Project Size Reducer Window Description](#).

- **Set Plugin Import Settings**

Sets import settings for files in the `Plugins` folder appropriately in bulk.

Please execute when changing Unity editor version or settings.

- **Import Extra Package**

Imports additional packages necessary when you want to reduce build size by excluding the `opencv_contrib` module. For details, please see [Tips for Each Platform](#).

- **Disable Use Unsafe Code**

Adds `OPENCV_DONT_USE_UNSAFE_CODE` to `ScriptCompilationDefines` in `Build Settings`. With this setting, OpenCV for Unity's internal code will not use unsafe code.

Enabling unsafe code often results in faster code, so we recommend keeping it enabled.

WebGL Settings

This section is for settings to improve performance during **WebGL** builds.

- **Enable SIMD Support**

Adds settings to enable **SIMD** (Single Instruction Multiple Data) instruction set to `PlayerSettings.WebGL.emscriptenArgs` to improve performance.

Please enable when aiming for significant speedup in WebGL builds.

- **Enable MultiThreads Support**

Enables `PlayerSettings.WebGL.threadsSupport` to enable multi-threading in WebGL.

Please enable when aiming for speedup in WebGL builds.

- **Enable wasm2023 Support**

Enables `PlayerSettings.WebGL.wasm2023` to enable the latest **WebAssembly 2023** support.

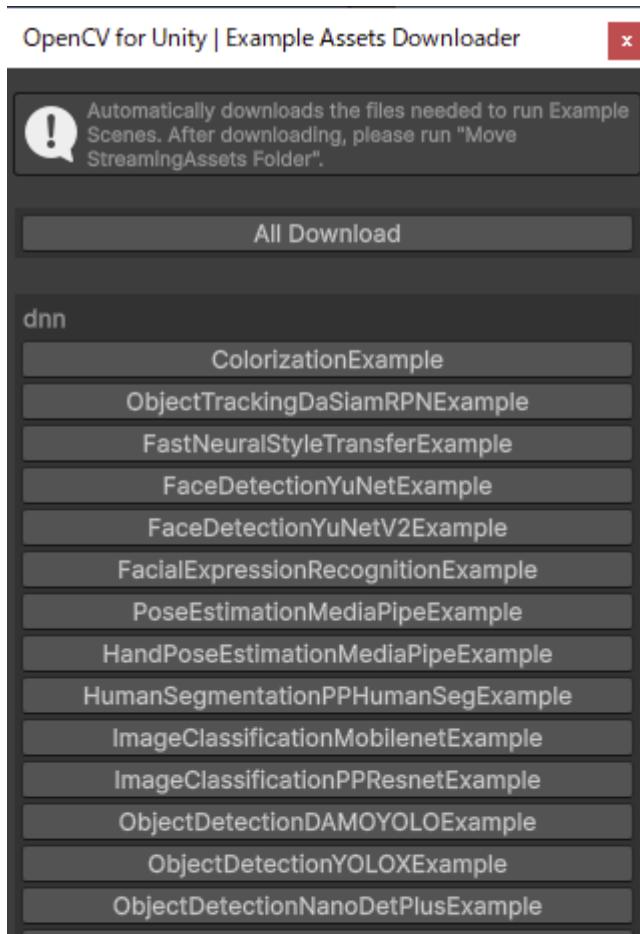
This setting automatically enables SIMD, WebAssembly native exceptions, BigInt, WebAssembly.Table, optimized data operations, and faster data type conversions.

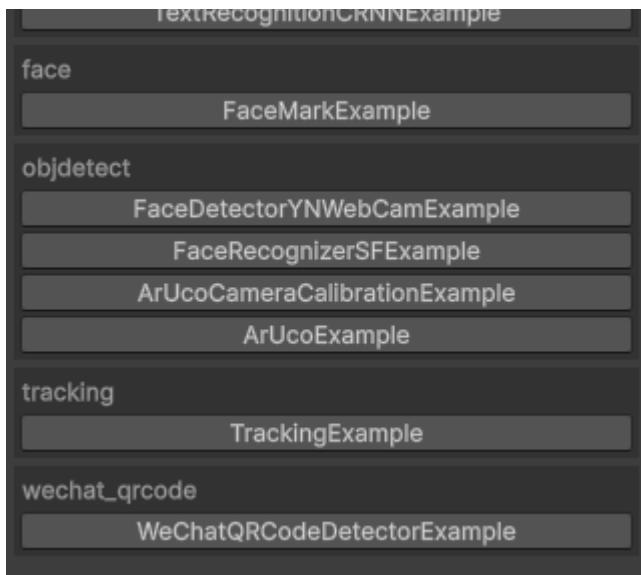
This setting is supported in Unity 2023.3 and later.

For details, please see [Unity Official Documentation](#).

Example Assets Downloader Window Description

Select the menu item `Tools > OpenCV for Unity > Open Example Assets Downloader` to open the `Example Assets Downloader` window. It can also be opened from the `Setup Tools` window.

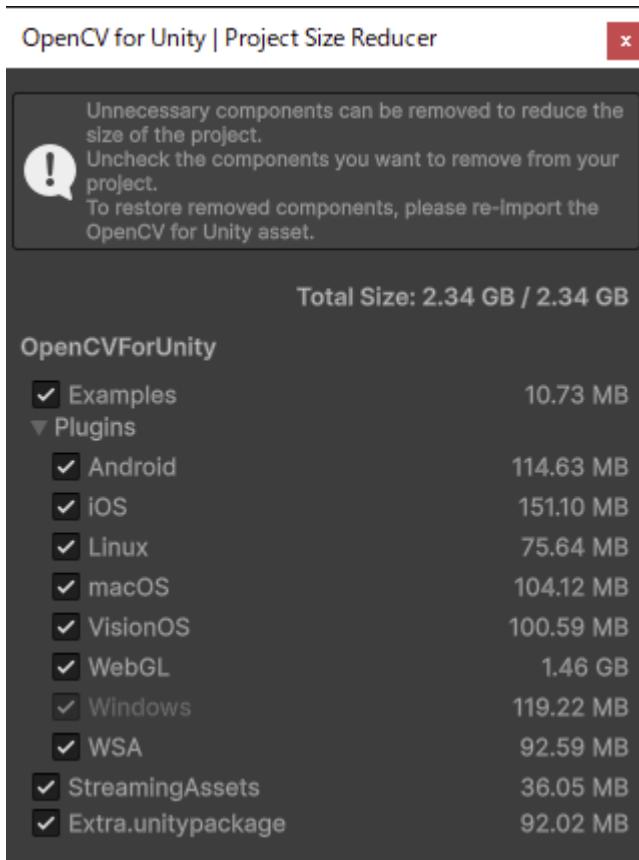


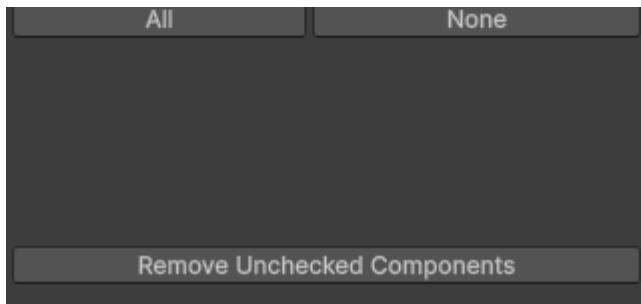


In this window, you can automatically download files necessary to run Example Scenes. The **All Download** button downloads files necessary for all Example Scenes. To download files necessary for specific scenes, click the respective Example Scene buttons. After downloading, please execute the **Move StreamingAssets Folder** button.

Project Size Reducer Window Description

Select the menu item **Tools > OpenCV for Unity > Open Project Size Reducer** to open the **Project Size Reducer** window. It can also be opened from the **Setup Tools** window.





In this window, you can remove unnecessary components to reduce project size. Uncheck the components you want to remove from the project and click the `Remove Unchecked Components` button. To restore deleted components, reimport the **OpenCV for Unity** asset.

Tips for Each Platform

- **Android**

- If you don't use the `opencv_contrib` module, you can reduce build size by using native plugin files that exclude the `opencv_contrib` module.
 - a. Click the `Import Extra Package` button to import the `Assets/OpenCVForUnity/Extra` folder.
 - b. Replace the `Assets/OpenCVForUnity/Plugins/Android/libs` folder with the `Assets/OpenCVForUnity/Extra/exclude_contrib/Android/libs` folder.
 - c. Click the `Set Plugin Import Settings` button.
 - d. Delete the `Assets/OpenCVForUnity/org/opencv_contrib` folder and `Assets/OpenCVForUnity/Examples/ContribModules` folder.

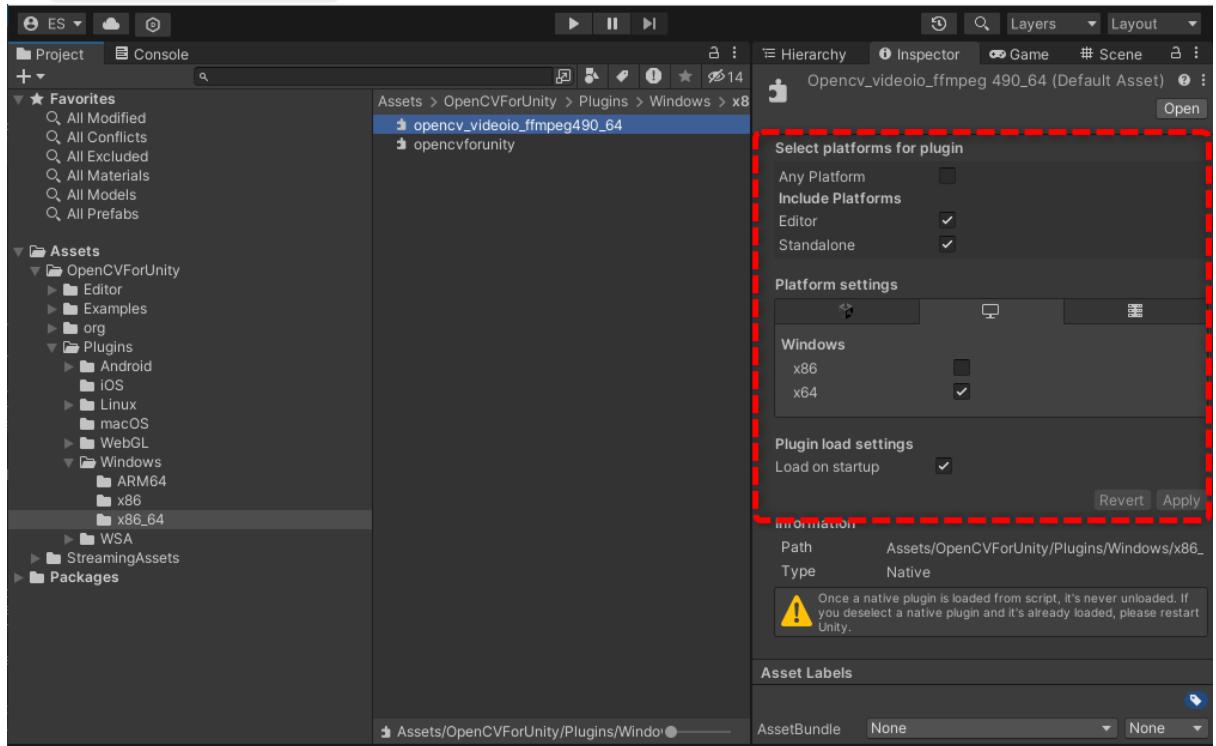
- **iOS**

- If you don't use the `opencv_contrib` module, you can reduce build size by using native plugin files that exclude the `opencv_contrib` module.
 - a. Click the `Import Extra Package` button to import the `Assets/OpenCVForUnity/Extra` folder.
 - b. Replace the `Assets/OpenCVForUnity/Plugins/iOS` folder with the `Assets/OpenCVForUnity/Extra/exclude_contrib/iOS` folder.
 - c. Click the `Set Plugin Import Settings` button.
 - d. Delete the `Assets/OpenCVForUnity/org/opencv_contrib` folder and `Assets/OpenCVForUnity/Examples/ContribModules` folder.

- **Windows**

- If you want to use more video formats by specifying the ffmpeg backend in `Video Capture (string filename)` or `VideoWriter` methods, setup is required.
 - a. Download **OpenCV for Windows Version 4.12.0** (<http://opencv.org/downloads.html>).

- b. Copy `opencv/build/bin/opencv_videoio_ffmpeg412_64.dll` to the `Assets/OpenCVForUnity/Plugins/Windows/x86_64` folder.
- c. Set the `Import Settings` of `opencv_videoio_ffmpeg412_64.dll` to the same settings as `opencvforunity.dll` in the same folder.



You can use ffmpeg as the backend by setting `Videoio.CAP_FFMPEG` to `apiPreference` as shown in the following code.

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());
```

• WebGL

- If you want to minimize the app's build size, try the following steps.

- a. `Player Settings` settings related to build size optimization:

```
Code Optimization : Size , il2cppCodeGeneration : Faster (smaller) builds ,
managedStrippingLevel : High , DecompressionFormat : Brotli .
```

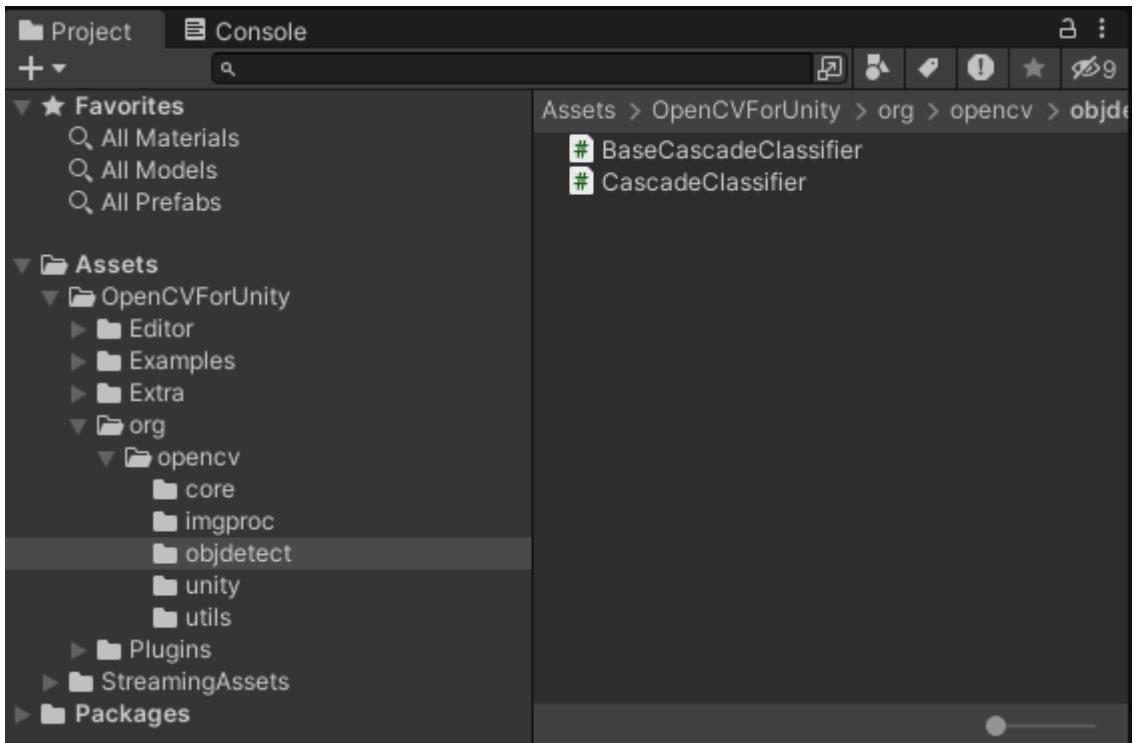
- b. Remove unused code from the `Assets/OpenCVForUnity/org` folder.

For example, the result when only the files necessary for `FaceDetectionExample` to work are left:

```

Code Optimization : Size , il2cppCodeGeneration : Faster (smaller) builds ,
managedStrippingLevel : High , DecompressionFormat : Brotli
→ Result: 6.31 MB (6,622,861 bytes) → Build folder only result 5.41 MB (5,681,838
bytes)

```



- **visionOS (beta)**

- On Vision Pro, even if you use `WebCamTexture`, you cannot obtain real-world camera images (pass-through images). Example Scenes using `WebCamTexture` will not work, so for applications that use camera functionality, you need to use Vision Pro-specific camera APIs.

How to use OpenCV Dynamic Link Library with customized build settings

Download **OpenCV 4.12.0** repository (git: [opencv](#), [opencv-contrib](#)).

- **Android**

- Change `self.cmake_vars['ANDROID_STL'] = 'c++_shared'` to `'c++_static'` in https://github.com/opencv/opencv/blob/4.x/platforms/android/build_sdk.py.
- 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

- iii. Copy the output file (`native\libs\arm64-v8a\libopencv_java4.so`) to `OpenCVForUnity\Plugins\Android\libs\arm64-v8a` folder.
Copy the output files (`native\libs\armeabi-v7a\libopencv_java4.so`) to `OpenCVForUnity\Plugins\Android\libs\armeabi-v7a` folder.
Copy the output files (`native\libs\x86_64\libopencv_java4.so`) to `OpenCVForUnity\Plugins\Android\libs\x86_64` folder.
- iv. Copy `OpenCVForUnity\Extra\dll_version\Android\libs` folder to `OpenCVForUnity\Plugins\Android\libs` folder.

- **iOS**

- i. Build the **iOS framework** with `"opencv/platforms/ios/build_framework.py"`.

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

- ii. Copy the output file (`opencv2.framework`) to `OpenCVForUnity\Plugins\iOS` folder.

- **Windows**

- i. Build the **OpenCV dynamic library**.

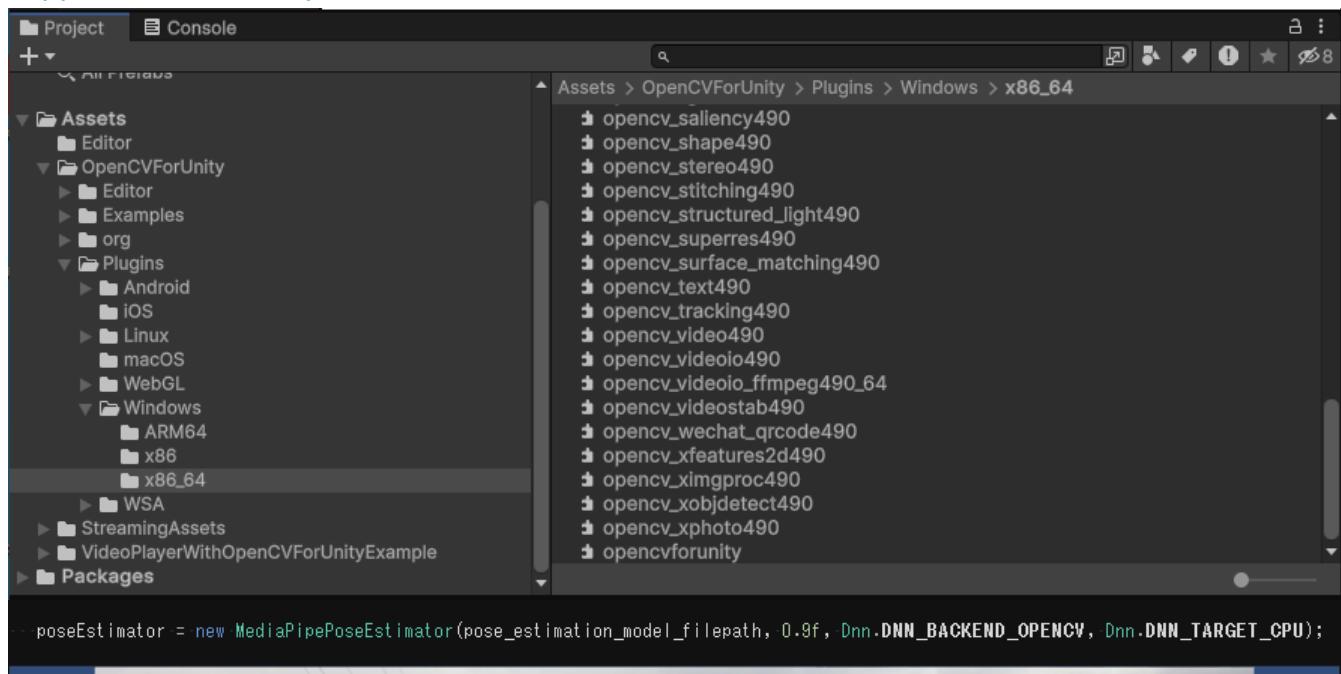
```
OPENCV_EXTRA_MODULES_PATH:PATH=C:/Users/xxxxx/opencv_contrib/modules BUILD_SHARED_LIBS:BOOL=ON
```

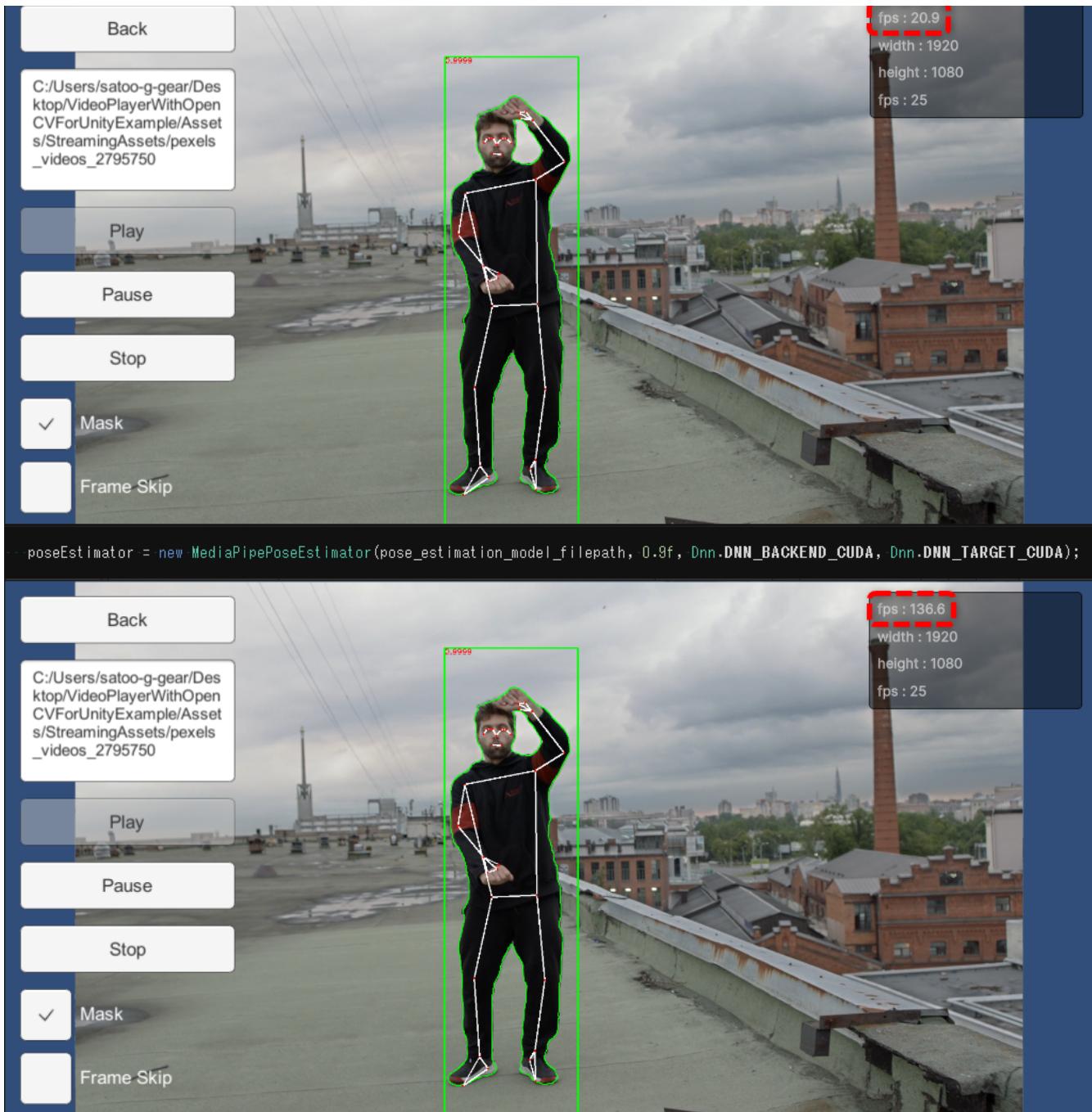
- ii. Copy `install\x64\vc16\bin` to `OpenCVForUnity\Plugins\Windows\x64` folder.

Copy `OpenCVForUnity\Extra\dll_version\Windows` to `OpenCVForUnity\Plugins\Windows` folder.

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**

- i. Build the **OpenCV library**.

```
CMAKE_OSX_ARCHITECTURESSTRING=x86_64
```

```
OPENCV_EXTRA_MODULES_PATHPATH=/Users/xxxxx/opencv_contrib/
modules
```

- ii. Copy the output files (`libopencv_*.4.12.0.dylib`) to `opencvforunity.bundle\Contents\MacOS` folder.

- iii. Relink `libopencv_*.4.12.0.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.4.12.0.dylib
```

```
install_name_tool -id @loader_path/libopencv_aruco.4.12.0.dylib
```

```

libopencv_aruco.4.12.0.dylib
install_name_tool -change @rpath/libopencv_calib3d.4.12.dylib @loader_path/
libopencv_calib3d.4.12.0.dylib libopencv_aruco.4.12.0.dylib
install_name_tool -change @rpath/libopencv_features2d.4.12.dylib
@loader_path/libopencv_features2d.4.12.0.dylib libopencv_aruco.4.12.0.dylib
install_name_tool -change @rpath/libopencv_flann.4.12.dylib @loader_path/
libopencv_flann.4.12.0.dylib libopencv_aruco.4.12.0.dylib
install_name_tool -change @rpath/libopencv_highgui.4.12.dylib @loader_path/
libopencv_highgui.4.12.0.dylib libopencv_aruco.4.12.0.dylib
install_name_tool -change @rpath/libopencv_videoio.4.12.dylib @loader_path/
libopencv_videoio.4.12.0.dylib libopencv_aruco.4.12.0.dylib
install_name_tool -change @rpath/libopencv_imgcodecs.4.12.dylib
@loader_path/libopencv_imgcodecs.4.12.0.dylib libopencv_aruco.4.12.0.dylib
install_name_tool -change @rpath/libopencv_imgproc.4.12.dylib
@loader_path/libopencv_imgproc.4.12.0.dylib libopencv_aruco.4.12.0.dylib
install_name_tool -change @rpath/libopencv_core.4.12.dylib @loader_path/
libopencv_core.4.12.0.dylib libopencv_aruco.4.12.0.dylib

```

- Linux

- i. Build the **OpenCV library**.
- ii. Rename output files (`libopencv_xxxxx.so.4.12.0`).

```

sudo apt-get install rpi
sed -i -e 's/.so.412/412.so/g' ./libopencv.so.4.12.0*
rename "s/.so.4.12.0/412.so/" libopencv.so.4.12.0*
find . -type f -name '_412.so' -print -exec patchelf --set-rpath '$ORIGIN' {} ;*

```

- iii. Copy `libopencv_xxxxx_412.so` to `OpenCVForUnity\Plugins\Linux\x86_64` folder.

- UWP

- i. Build the **OpenCV library**.

```

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

```

- ii. Copy `install/WS/10.0/ARM/ARM/vc16/bin` folder to

`Assets/OpenCVForUnity/Plugins/WSA/UWP/ARM` folder. Copy

`install/WS/10.0/x64/x64/vc16/bin` folder to `Assets/OpenCVForUnity/Plugins/WSA/UWP/x64` folder. Copy `install/WS/10.0/x86/x86/vc16/bin` folder to `Assets/OpenCVForUnity/Plugins/WSA/UWP/x86` folder.

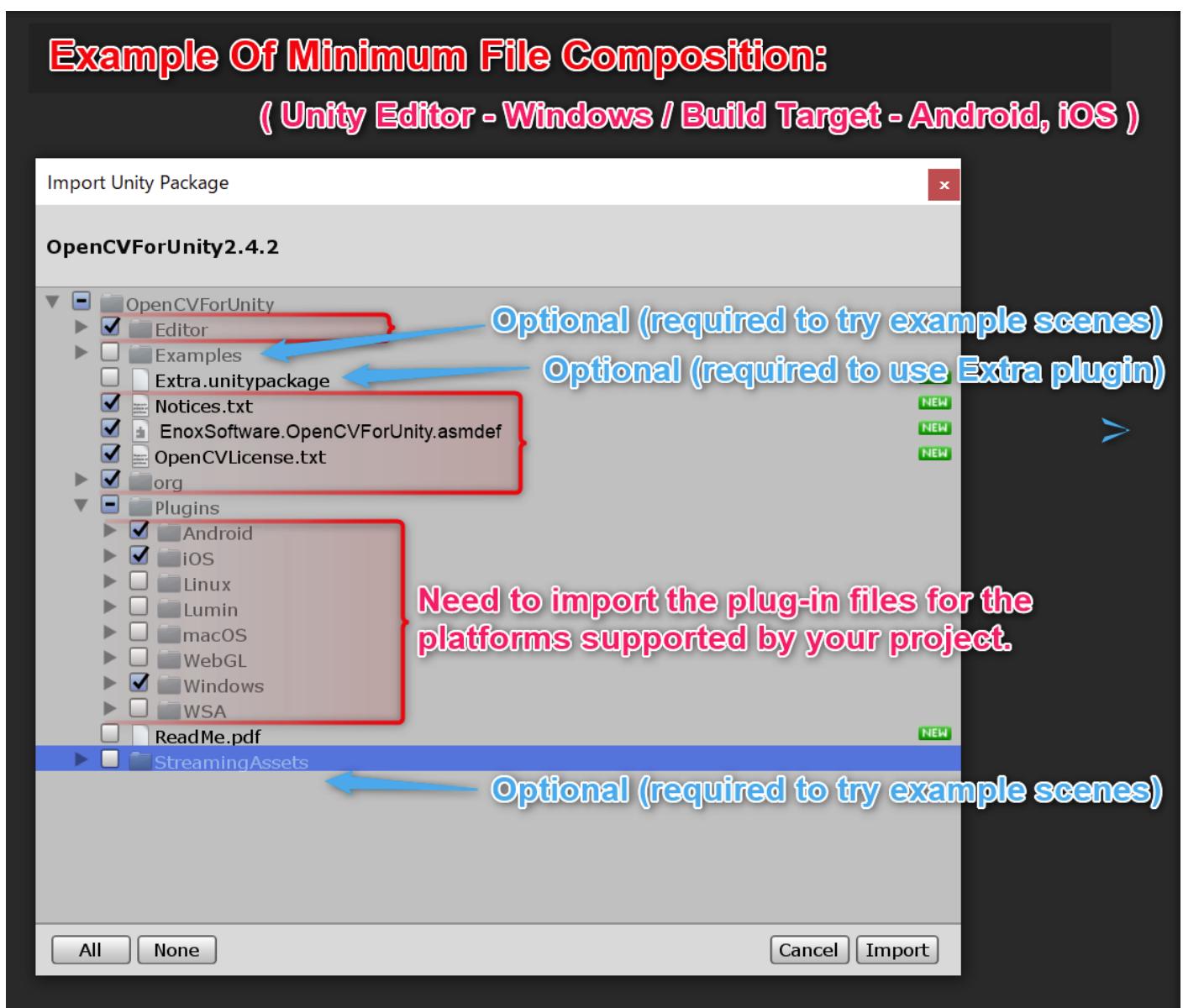
Q & A

Q1. The asset package size is large. Is there a way to reduce it?

A1. Using the `Project Size Reducer`, you can easily reduce unnecessary files in the `Assets/OpenCVForUnity` folder. Particularly, by removing plugin files for platforms not planned to be supported in your development project, you can significantly reduce the project size. For details, please see [Example Assets Downloader Window Description](#).

Q2. What is the minimum file configuration required for the asset to work?

A2. You don't need to import all files for the asset to work. If you don't need to try Example Scenes, the minimum file configuration required is as follows.



Q3. How can I learn the details of OpenCV methods and arguments?

A3. For details on method arguments, please refer to [OpenCV Official Documentation](#) and [OpenCV Tutorials](#).

Q4. How can I convert Mat class operators defined in C++?

A4. Regarding the conversion method of `Mat` class operators defined in **C++**. For details, please see this page. ([Way to translation of Mat class operators defined in C++](#))

Q5. When running Example Scenes, D11NotFoundException: opencvforunity appears in the console.

A5. It seems the plugin is not loaded correctly. Please check the setup procedure.

Q6. When running Example Scenes, ArgumentException: The output Mat object has to be of the same appears in the console.

A6. After plugin setup, restarting the **Unity Editor** may make the plugin work properly.

Q7. When running Example Scenes, Level 'Texture2DtoMatExample' (-1) could not be loaded because appears in the console.

A7. Click the `Add Example Scenes in Build` button in the `Setup Tools` window to add all Example Scene files in the `Examples` folder to `Scenes In Build` in `Build Settings`.

Q8. How can I use SIFT and SURF algorithms?

A8. The native libraries included in **OpenCV for Unity** are built with the `OPENCV_ENABLE_NONFREE` flag disabled. To use **SIFT** and **SURF** algorithms, please rebuild the OpenCV library with `OPENCV_ENABLE_NONFREE` enabled. For details, please see [How to use OpenCV Dynamic Link Library with customized build settings](#).

Q9. How to catch error codes that occur on the native side of OpenCV (CVException handling)

A9. To display error codes from the native side of OpenCV, wrap your code with

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

```
Utils.setDebugMode(true);  
// your OpenCV calls here  
Utils.setDebugMode(false);
```

For details, please see this page. ([How to catch native OpenCV's errors code \(CVException handling\)](#))

Version Changes

3.0.0 [Common] Updated OpenCV to version 4.12.0. [Common] Updated FpsMonitor to version 1.0.4. [visionOS] Updated platform support for visionOS on Unity 2022.3.18f1 and later (beta). [Android] Updated native library libopencvforunity.so for Android to be compatible with 16KB page size. [Common] Added SafeDrawFrameAxes method to OpenCVARUtils. [Common] Added GetFilePathCoroutine, GetFilePathAsync, and GetFilePathTaskAsync methods to OpenCEnv. [Common] Added IsDebugMode and IsThrowException methods to OpenCVDebug. [Common] Added TextureToMatAsync, RenderTextureToMatAsync, and MatToRenderTextureAsync methods to OpenCVMatUtils. [Common] Added OpenCVForUnityProjectSizeReducerWindow in EditorWindow. [Common] Added a feature to skip unnecessary downloads in OpenCVForUnityExampleAssetsDownloaderWindow. [Common] Added ImshowDNNBlob method to DebugMat. [Common] Added WeChatQRCodeDetectorImageExample and WeChatQRCodeDetectorExample. [Common] Added HomographyToFindAKnownObjectExample. [Common] Added ObjectDetectionDAMOYOLOExample. [Common] Removed FaceDetectionResnetSSDEExample due to incompatibility with the latest OpenCV. [Common] Changed namespaces under org/unity from UnityUtils to UnityIntegration, split Utils class into function-specific classes, and marked old classes as deprecated. [Common] Changed Operator in Mat class to deprecated.

2.6.6 [Common] Changed The FpsMonitor display system from IMGUI to uGUI. [Common] Added support for the new Input System (UnityEngine.InputSystem).

2.6.5 [Common] Add ARHelper component class. [Common] Updated ArUcoExample, ArUcoImageExample, HandPoseEstimationMediaPipeExample and PostEstimationMediaPipeExample using the ARHelper class. [Common] Refactored AsyncGPUReadback2MatHelper, Image2MatHelper, UnityVideoPlayer2MatHelper, VideoCapture2MatHelper, WebCamTexture2MatAsyncGPUHelper and WebCamTexture2MatHelper. [Common] Added method to convert OpenCV basic classes to Unity basic structs.

2.6.4 [Common] Added various data access methods to the Mat class, including ptr(), at(),

AsSpan(), and AsSpanRowRange(). [Common] Updated get() and set() methods in the Mat class to support Span and generics, and updated MatUtils.copyFromMat() and MatUtils.copyToMat() to support Span. [Common] Added structs and ValueTuples corresponding to the data formats of basic OpenCV classes, and created overloads for methods that use these basic classes as arguments, allowing the use of structs and ValueTuples instead. This update simplifies code, enables more efficient memory usage, and improves data access speed. [Common] Added Utils.getFilePathAsyncTask() and Utils.getMultipleFilePathsAsyncTask() methods. It is an asynchronous Task return version of the existing Utils.getFilePathsAsync() method. It can be seamlessly integrated with other asynchronous code. [Common] Added Utils.texture2DToMatRaw() and Utils.matToTexture2DRaw() methods. [Common] Added methods for conversion between Mat and RenderTexture using ComputeShader (Utils.matToRenderTexture(), Utils.renderTextureToMat()). [Common] Added the WebCamTexture2MatAsyncGPUHelper class. [Common] Updated the DebugMatUtils class to support displaying Texture2D and RenderTexture. [Common] Added a face blurring feature to FaceDetectorYNEExample. [iOS] Fixed an issue where AddToEmbeddedBinaries was processed redundantly with each incremental build.

2.6.3 [Common]Fixed CountFingersExample. [Windows][UWP]Fixed native library build settings for ARM64 architecture.

2.6.2 [Common]Changed the minimum supported version to Unity2021.3.35f1. [Common]Added AsyncGPUReadback2MatHelper. [Common]Updated MultiSource2MatHelperExample. [Common]Separated the examples using the Built-in Render Pipeline and Scriptable Render Pipeline. [Common]Fixed a memory leak in Converters.cs.

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 TransformECCExample, 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 MaskRCNNExample. [WebGL] Added Unity2019.1 or later support.

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.