OpenCV ObjectDetector 1.1.4

iOS & **Android** support

Win & **Mac** Standalone support

Support for preview in the **Editor**

Work with Unity Free & Pro

System Requirements

Build Win Standalone & Preview Editor : Windows7 or later

Build Mac Standalone & Preview Editor : OSX 10.8 or later

"OpenCV ObjectDetector" can detect(Sync or Async) an object from Texture2D using OpenCV.

- You can get a processing result of detectMultiScale() of OpenCV using haar cascade file that you specified.
- Object detection parameters (same as the parameters of detectMultiScale()) can be set in JSON format, You can get in JSON format Object detection result.

Please download Demo Application for Android and watch tutorial video.

Version changes

1.1.4 [Common]Add OpenCVObjectDetectorMenuItem.cs.(This script set plugin import settings automatically from MenuItem.) [iOS]Move "OpenCVObjectDetector/ iOSforXcode/iOS_BuildPostprocessor.cs" to "OpenCVObjectDetector/Editor"folder.

1.1.3 [Common]Update to OpenCV2.4.11

1.1.2 [Common]Divide asset for Unity4 and Unity5.

1.1.1 [Common]Support for Unity5

1.1.0 [Common]Update to OpenCV2.4.10

1.0.9 [iOS]Support for arm64 build target.(Unity 4.6.1p3 or higher)

1.0.8 [Android]Support for x86 build target.(Unity 4.6 or higher)

1.0.7 [Common]Update SampleScene(Process of converting results of object detection to the 3D position).

1.0.6 [Common]Support for preview in the Editor.(Pro only) [Common]Support for Win & Mac Standalone.(Pro only) [Android]Change of location of the cascade file.Changed to use "Aseets/StreamingAssets/" folder. [iOS] Add the cascade file to Xcode project is no longer required.Changed to use"Aseets/StreamingAssets/" folder.

1.0.4 [iOS]fix library(libjpeg,libpng) version coflicts.

1.0.3 update ReadMe.pdf

1.0.2 [Common]Update to OpenCV2.4.9.[Common]Support LBP cascade file. [Android]opencv library – 2.4.8.jar is no longer required.[iOS] Link "libc++.dylib" to Xcode project is no longer required.

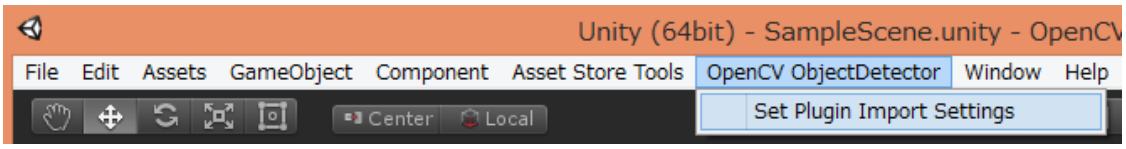
1.0.1 Remove unnecessary files.

1.0.0 Initial version

In Unity4 and Unity5 the different files to be imported. When you update the project that you made in Unity4 to Unity5, please import again this asset in Unity5.

Quick setup procedure to run the sample scene

1. If Unity4, Move "OpenCVObjectDetector/Plugins/"folder to "Assets/"folder.
2. If Unity5,Select MenuItem[OpenCV ObjectDetector/Set Plugin Import Settings].

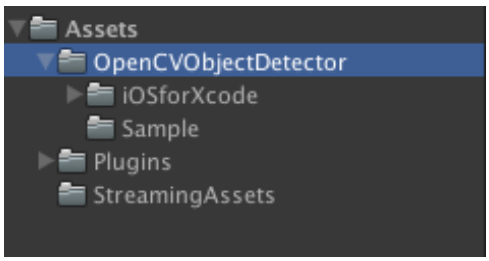


3. Move "OpenCVObjectDetector/StreamingAssets/"folder to "Assets/"folder.
4. Please set [PlayerSettings]-[Resolution and Presentation]-[Orientation]-[Default Orientation : Portrait] when you build the sample scene.
5. Add all of the "***.unity" in the "OpenCVObjectDetector" folder to [Build Settings] – [Scene In Build].

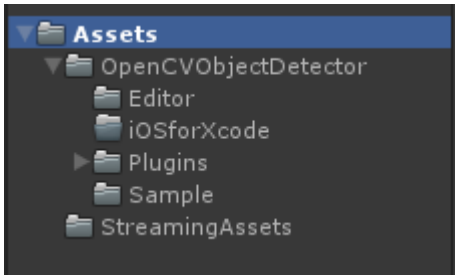
※Inspector Setting of "opencv2.framework" and "opencvobjectdetector.bundle" might have been reset at the time of import. In that case, re-setup is required.

Screenshot after the setup

Unity4



Unity5



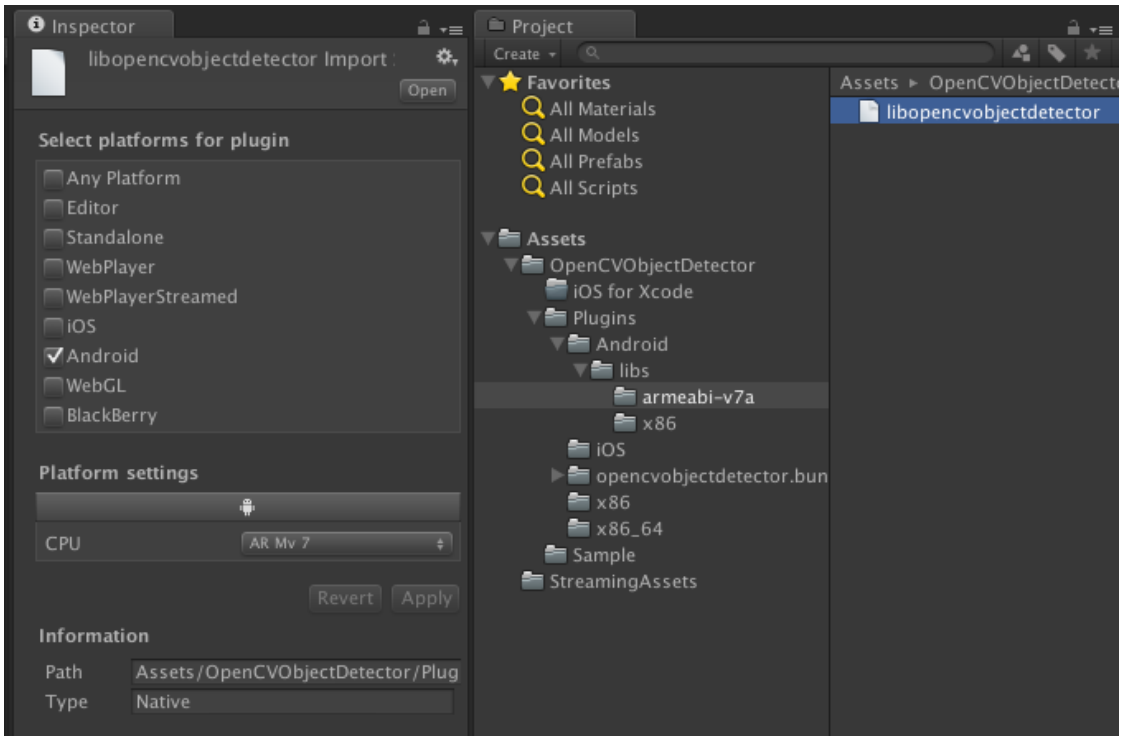
Android Setup

Unity4

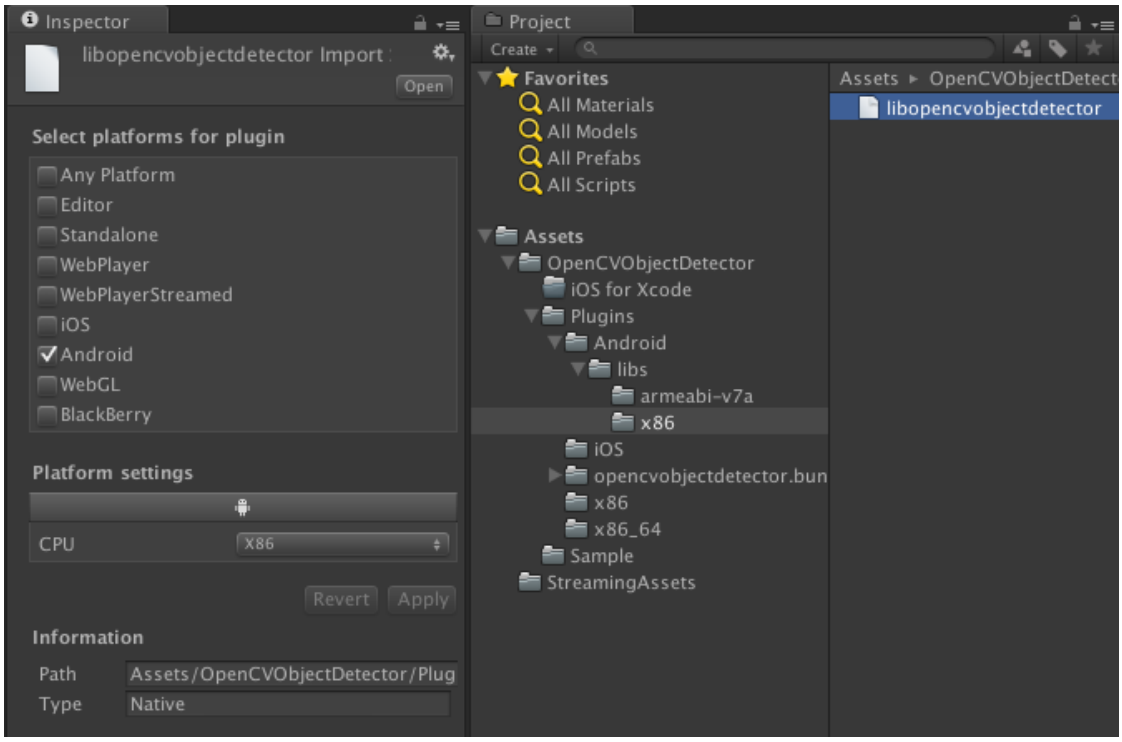
- Copy from "OpenCVObjectDetector/Plugins/Android/" to "Assets/Plugins/Android/" folder.

Unity5

- "OpenCVObjectDetector/Plugins/Android/opencvobjectdetector.jar" – Select platform Android in Inspector.
- "OpenCVObjectDetector/Plugins/libs/armeabi-v7a/*.so" – Select platform Android and CPU ARMv7 in Inspector.



- "OpenCVObjectDetector/Plugins/libs/x86/*.so" – Select platform Android and CPU x86 in Inspector.

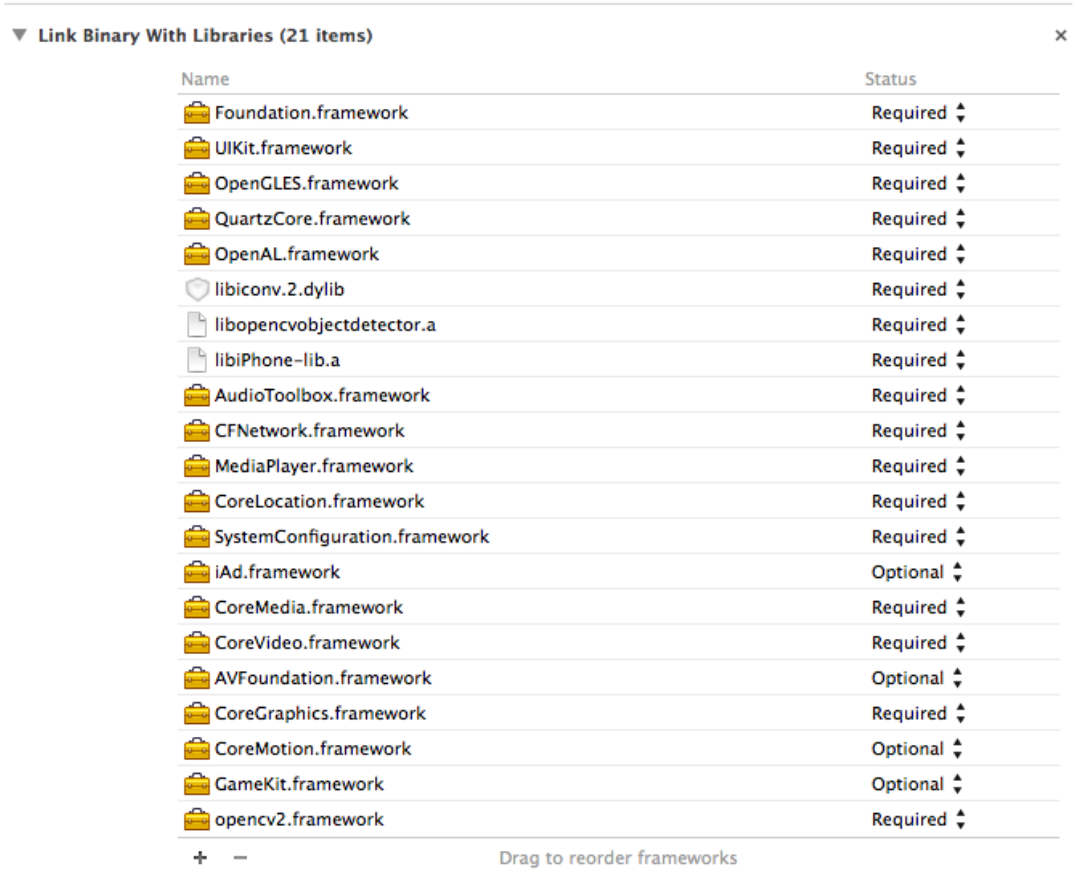


- Put the cascade file that you want to use for object detection in the "Aseets/StreamingAssets/".

iOS Setup

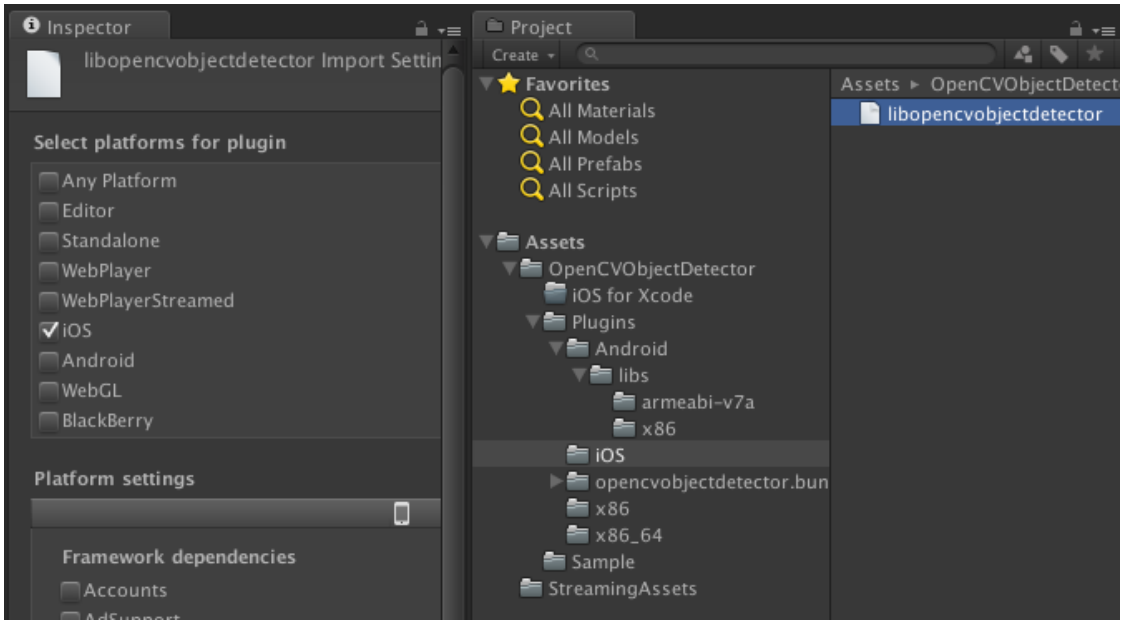
Unity4

- Copy from "OpenCVObjectDetector/Plugins/iOS/" to "Assets/Plugin/iOS/" folder.
- Link "OpenCVObjectDetector/iOS for Xcode/opencv2.framework" to Xcode project. (in Xcode project. Build Phases > Link Binary with Libraries > Add opencv2.framework . recommend to use PostprocessBuildPlayer.)

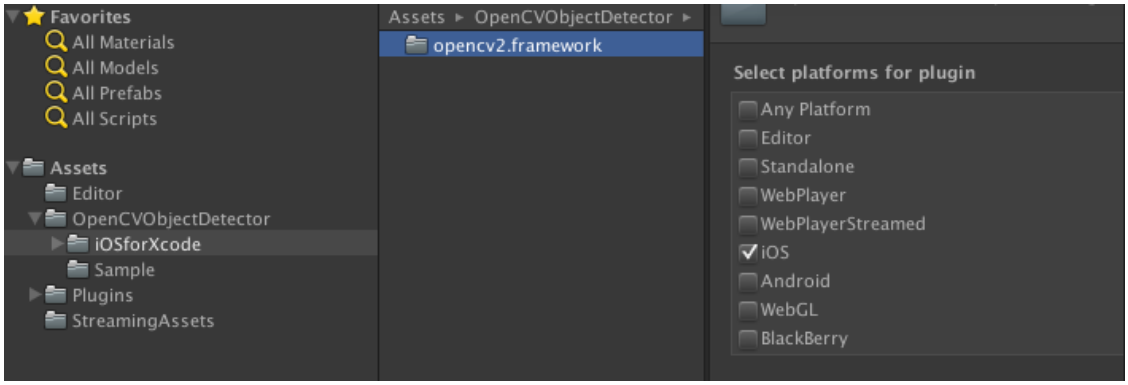


Unity5

- "OpenCVObjectDetector/Plugins/iOS/libopencvobjectdetector.a" – Select platform iOS in Inspector.



- "OpenCVObjectDetector/iOSforXcode/opencv2.framework" – Select platform iOS in Inspector.



- Put the cascade file that you want to use for object detection in the "Aseets/StreamingAssets/".

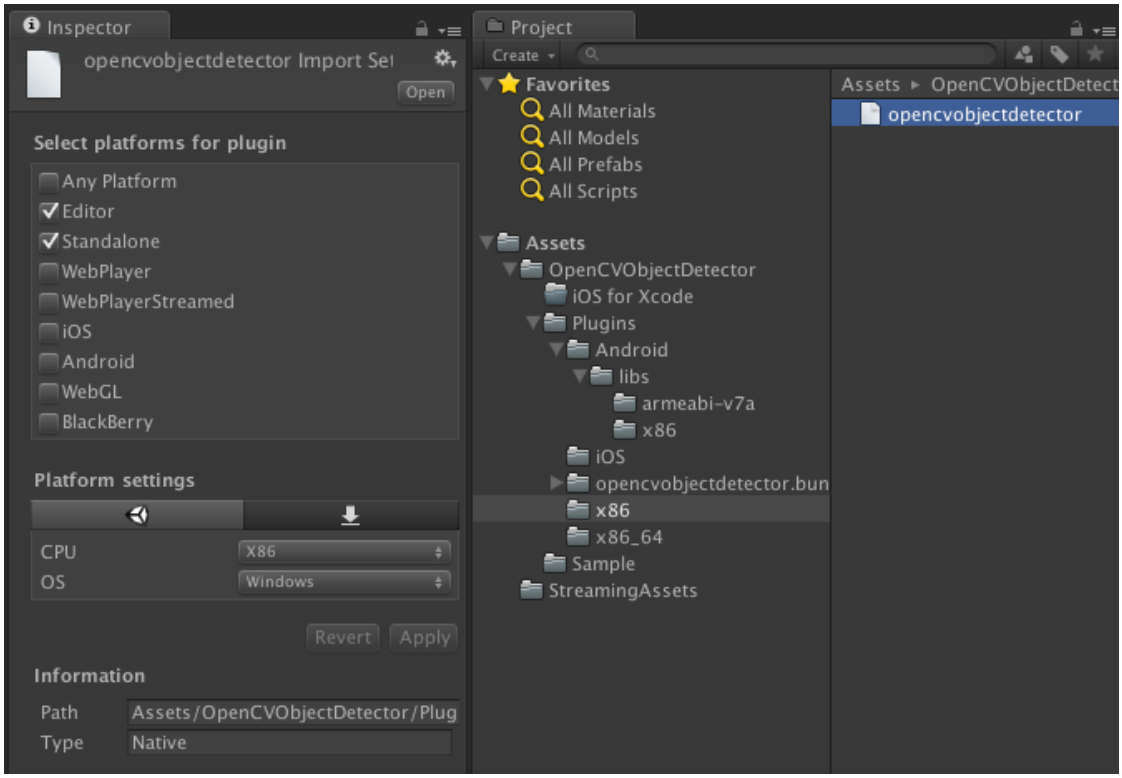
Win Standalone Setup

Unity4

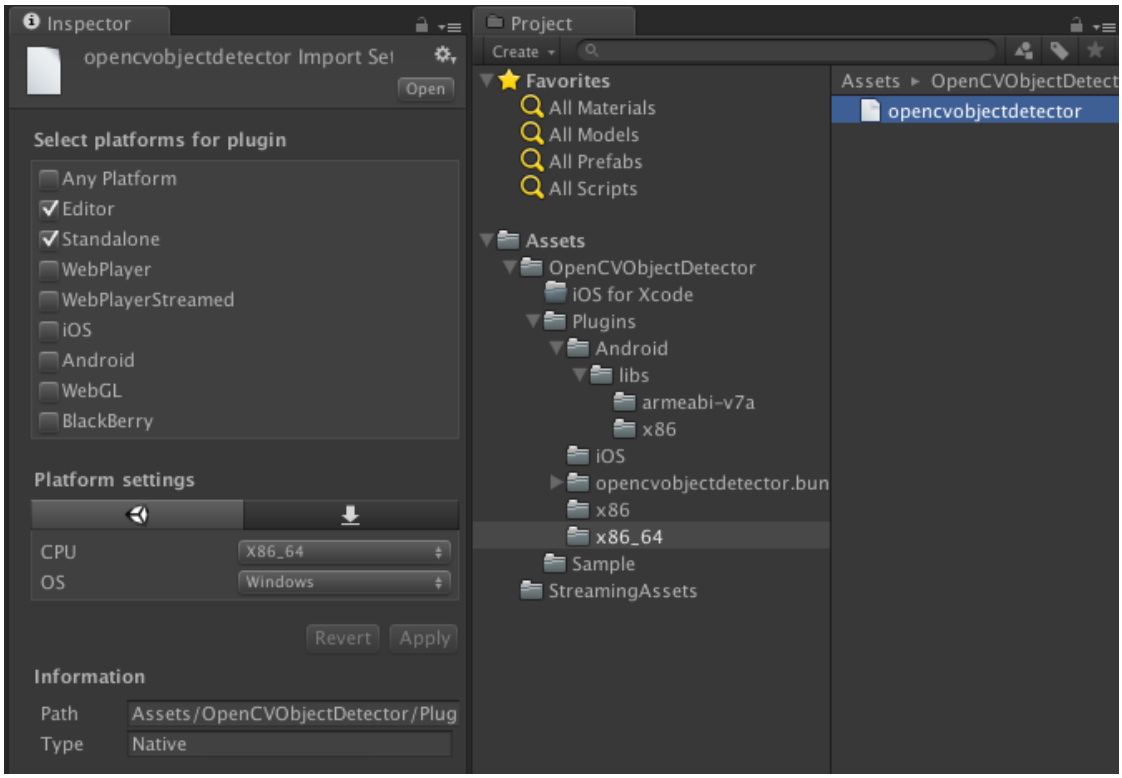
- Copy from "OpenCVObjectDetector/Plugins/x86/" to "Assets/Plugins/x86/" folder.
- Copy from "OpenCVObjectDetector/Plugins/x86_64/" to "Assets/Plugins/x86_64/" folder.

Unity5

- "OpenCVObjectDetector/Plugins/x86/opencvobjectdetector.dll" – Select platform Editor,Standalone and CPU x86 and OS Windows in Inspector.



- "OpenCVObjectDetector/Plugins/x86_64/opencvobjectdetector.dll" – Select platform Editor,Standalone and CPU x86_64 and OS Windows in Inspector.



- Put the cascade file that you want to use for object detection in the "Aseets/StreamingAssets/".

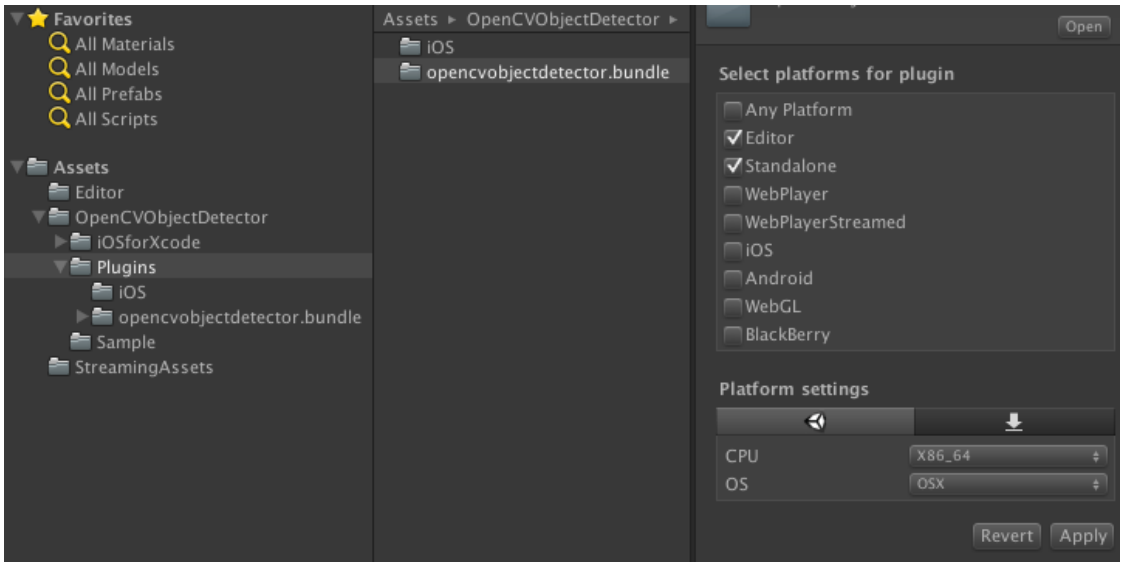
Mac Standalone Setup

Unity4

- Copy from "OpenCVObjectDetector/Plugins/opencvobjectdetector.bundle" to "Assets/Plugins/" folder.

Unity5

- "OpenCVObjectDetector/Plugins/opencvobjectdetector.bundle" – Select platform Editor,Standalone and CPU x86_64 and OS OSX in Inspector.



- Put the cascade file that you want to use for object detection in the "Aseets/StreamingAssets/".

Q & A

Q1.

"DllNotFoundException: opencvobjectdetector" is displayed on the console when run the sample scene.

A1.

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

Detect param example (JSON format)

```
{
   "filename":"haarcascade_frontalface_alt",    //haar cascade filename
   "scaleFactor":1.1, //Please refer to OpenCV cvHaarDetectObjects() arg.
   "minNeighbors":2, // Please refer to OpenCV cvHaarDetectObjects() arg.
   "flags":2,    // Please refer to OpenCV cvHaarDetectObjects() arg.
   "minWidth":80,    // Please refer to OpenCV cvHaarDetectObjects() arg.
   "minHeight":80,    // Please refer to OpenCV cvHaarDetectObjects() arg.

   "flipCode":0,    //(optional) flip the image in Detect. Please refer to OpenCV cv::flip arg.

   "rects":[    //(optional) Ranges of detection in Texture2D. To set when you want to detect part of the Texture2D. Texture2D is bottom-left origin.

      {
         "id":0,    // (optional)Id identify the detection range.default 0.
         "x":10,
         "y":10,
         "width":200,
         "height":300
      },
      {
         "id":1,    //(optional) Id identify the detection range.default 0.
         "x":200,
         "y":210,
         "width":150,
         "height":150
      }
   ]
}
```

Detect result example (JSON format)

```
{
   "haarcascade_frontalface_alt":[    //cascade filename that was used to detect.
      {
         "id":0,    //detection range id that you set in Detect param.
         "x":20,
         "y":35,
         "width":179,
         "height":179
      },
      {
         "id":1,    //detection range id that you set in Detect param.
         "x":211,
         "y":200,
         "width":100,
         "height":95
      }
   ]
}
```

IMPORTANT: READ BEFORE DOWNLOADING, COPYING, INSTALLING OR USING.

By downloading, copying, installing or using the software you agree to this license.
If you do not agree to this license, do not download, install,
copy or use the software.

License Agreement
For Open Source Computer Vision Library

Redistribution and use in source and binary forms, with or without modification,
are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice,
  this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice,
  this list of conditions and the following disclaimer in the documentation
  and/or other materials provided with the distribution.

* The name of the copyright holders may not be used to endorse or promote products
  derived from this software without specific prior written permission.

This software is provided by the copyright holders and contributors "as is" and
any express or implied warranties, including, but not limited to, the implied
warranties of merchantability and fitness for a particular purpose are disclaimed.
In no event shall the Intel Corporation or contributors be liable for any direct,
indirect, incidental, special, exemplary, or consequential damages
(including, but not limited to, procurement of substitute goods or services;
loss of use, data, or profits; or business interruption) however caused
and on any theory of liability, whether in contract, strict liability,

or tort (including negligence or otherwise) arising in any way out of
the use of this software, even if advised of the possibility of such damage.