OpenCV ObjectDetector

iOS & Android support

Work with Unity Free & Pro

OpenCV ObjectDetector can be the object detection(Sync or Async) from Texture2D by 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 read this guide and take a look at the included Demo Scene to see how to do this.Demo Application for Android.

Version changes

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

Android Setup

- Copy from "OpenCVObjectDetector/Plugins/Android/" to "Assets/Plugins/Android/" folder.
- Put the haar cascade file that you want to use for object detection in the "Plugins/Android/assets/"folder.

iOS Setup

- Copy from "OpenCVObjectDetector/Plugins/iOS/" to "Assets/Plugin/iOS/" folder.
- Add the haar cascade file that you want to use for object detection to Xcode project. (in Xcode project. Build Phases > Copy Bundle Resources > Add file. recommend to use PostprocessBuildPlayer.)
- 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.)

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.