StarFive
赛昉科技

使用昉·星光 2进行通用物体识别

应用指南

版本：1.0

日期：2023/06/19

Doc ID：VisionFive 2-ANCH-016

法律声明

阅读本文件前的重要法律告知。

版权注释

本文档中的说明均基于“视为正确”提供，可能包含部分错误。内容可能因产品开发而定期更新或修订。上海赛昉科技有限公司（以下简称“赛昉科技”）保留对本协议中的任何内容进行更改的权利，恕不另行通知。

赛昉科技明确否认任何形式的担保、解释和条件，无论是明示的还是默示的，包括但不限于适销性、特定用途适用性和非侵权的担保或条件。

赛昉科技无需承担因应用或使用任何产品或电路而产生的任何责任，并明确表示无需承担任何及所有连带责任，包括但不限于间接、偶然、特殊、惩戒性或由此造成的损害。

联系我们：

地址： 浦东新区盛夏路61弄张润大厦2号楼502，上海市，201203，中国

网站： http://www.starfivetech.com

邮箱：sales@starfivetech.com （销售） support@starfivetech.com（支持）

目录

表格清单

插图清单

前言

关于本指南和技术支持信息

关于本手册

本物体识别系列应用说明提供使用昉·星光 2进行多种物体识别的步骤，应用所使用的视觉框架基于OpenCV，赛昉科技对其进行了昉·星光 2的平台适配与底层GPU加速调优。

本文中的应用包括：

- 用于通用物体识别 Python及C++语言应用，含三个应用对应三种通用物体识别模型：
    1. YOLO-V3 模型
    2. 由 PyTorch 格式转换为 ONNX 格式的 YOLO-V5 模型
    3. MobileNet-SSD 模型

修订历史

表 0-1 修订历史

版本	发布日期	修订
1.0	2023/06/19	首次发布。

注释和注意事项

本指南中可能会出现以下注释和注意事项：

- **提示：**
  建议如何在某个主题或步骤中应用信息。
- **注：**
  解释某个特例或阐释一个重要的点。
- **重要：**
  指出与某个主题或步骤有关的重要信息。
- **警告：**
  表明某个操作或步骤可能会导致数据丢失、安全问题或性能问题。
- **警告：**
  表明某个操作或步骤可能导致物理伤害或硬件损坏。

1. 应用简介

本文中的应用包括：

- 用于通用物体识别 Python及C++语言应用，含三个应用对应三种通用物体识别模型：
    1. YOLO-V3 模型
    2. 由 PyTorch 格式转换为 ONNX 格式的 YOLO-V5 模型
    3. MobileNet-SSD 模型

以上应用属于赛昉科技实现的目标识别系列应用，更多信息请参见目标识别系列应用 (第 16页)。

2. 准备

在执行演示程序之前，务必确认已准备好以下项目：

2.1. 运行环境要求

该演示运行环境要求如下：

- Linux内核版本：Linux 5.15
- 操作系统：Debian 12
- 硬件版本：昉·星光 2
- SoC：昉·惊鸿7110

2.2. 准备硬件

在执行演示程序之前，请务必准备以下硬件：

表 2-1 硬件准备

类型	M/O*	项目	注释
通用	M	昉·星光 2 单板计算机	-
通用	M	• 容量不低于32 GB的Micro-SD卡 • Micro-SD卡读卡器 • 计算机（Windows/Mac OS/Linux） • USB转串口转换器（3.3 V I/O，带线） • 以太网电缆 • 电源适配器（5 V/ 3 A） • USB Type-C数据线	上述项目用于将Debian OS烧录到Micro-SD上。
物体识别应用	M	显示器及配件： • 一个HDMI显示器 • 一根HDMI连接线	-
	M	摄像头及配件： • 一个USB摄像头 • 一个IMX219 MIPI摄像头	适配的USB摄像头列表可参阅：JH110 AVL
	M	USB键盘	用于在 Debian 系统上操作终端
	O	USB鼠标	用于在 Debian 系统上操作终端

注：
*: M：必须。O：可选

2.2.1. 连接硬件

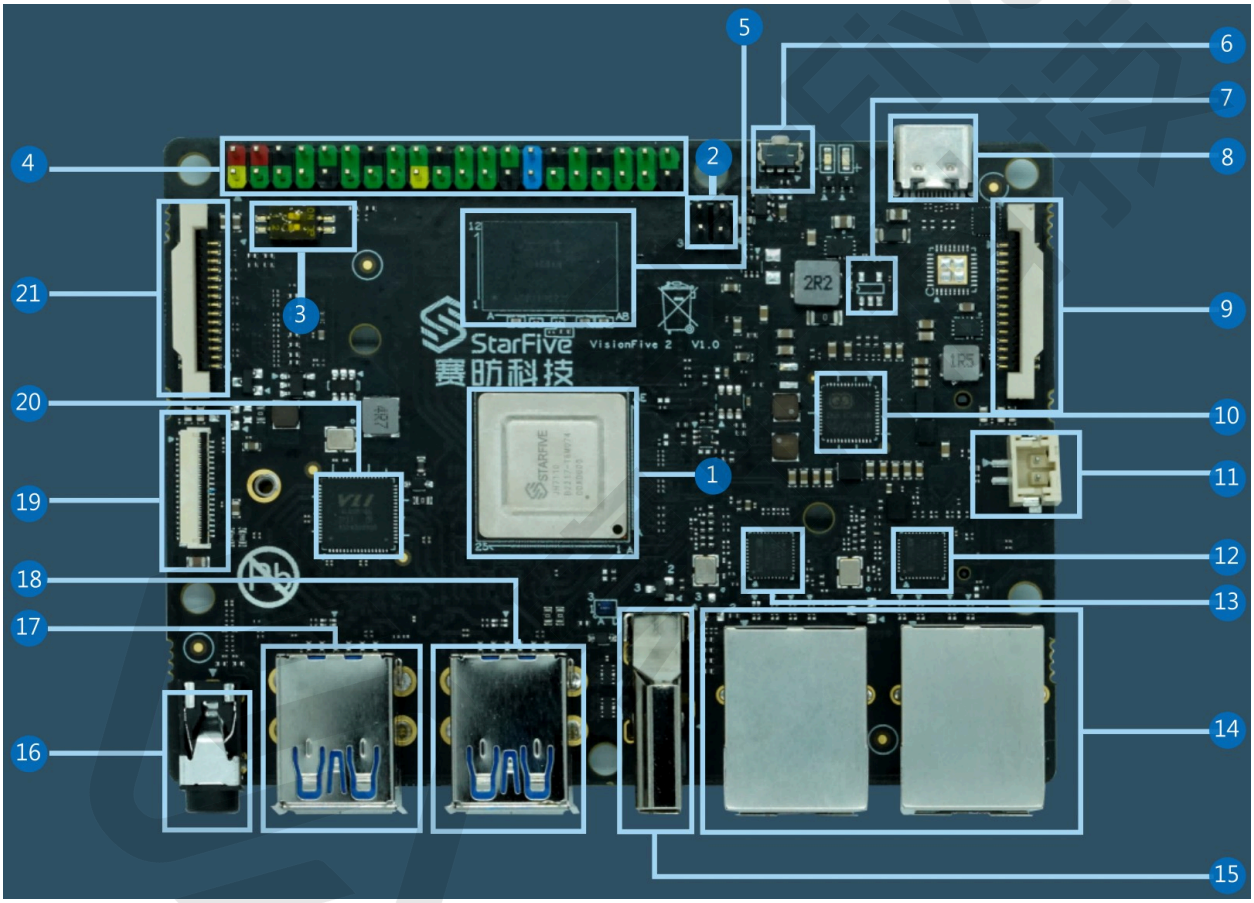
请参照如下昉·星光 2顶部视图，将外设、附件与电源等连接到昉·星光 2 的对应接口上：

1. 将烧录好Debian系统镜像的 MicroSD 卡连接到背面的Micro-SD卡槽上。

   注：
   按照《昉·星光 2单板计算机快速参考手册》中的“将*OS*烧录到*Micro-SD*”章节，将Debian OS烧录到Micro-SD卡上。请确保您下载的Debian镜像为最新版本，以镜像所在文件夹创建时间为准。

2. 将网线连接到编号 14 的任一 RJ45 以太网接口上。
3. 将显示器通过 HDMI 连接线连接到编号 15 的 HDMI 连接器上。
4. 将 IMX219 MIPI 摄像头连接到编号 9 的 MIPI CSI 连接器上。摄像头底座若有用于固定的排针座（不连接实际电路），可将其固定在编号 4 的 40 pin 空闲插槽上。（可选）
5. 将 USB 摄像头、键盘、鼠标连接到编号 17 与 18 的 USB 接口上。
6. 将 USB 电源适配器通过 USB-C 线连接到编号 8 的 USB-C 接口上。（必须在执行上述1~4步之后进行）

图 2-1 昉·星光 2顶部视图



2.3. 准备软件

确认按照以下步骤进行操作:

注：
请确保您使用的是最新的Debian镜像。

1. 登录Debian并确保昉·星光 2已联网。有关详细说明，请参阅《昉·星光 *2*单板计算机快速参考手册》中“通过以太网使用*SSH*”或“使用*USB*转串口转换器”章节。

   注：
   按照《昉·星光 *2*单板计算机快速参考手册》中的“将*OS*烧录到*Micro-SD*”章节，将Debian OS烧录到Micro-SD卡上。请确保您下载的Debian镜像为最新版本，以镜像所在文件夹创建时间为准。

2. 登录Debian并确保昉·星光 2已联网。借助键盘、鼠标与HDMI显示器，在 Debian 系统上登录，有关详细说明，请参阅《昉·星光2单板计算机快速参考手册》中“通过HDMI使用Xfce桌面环境登录” 章节。

3. 在Debian上扩展分区，请参见《昉·星光2单板计算机快速参考手册》中“扩展分区”章节。

4. 在昉·星光 2 Debian上执行如下命令安装 `StarFive Packages` 及其依赖，安装完成需要1-3小时：

   ```
   https://
   github.com/starfive-tech/Debian/releases/download/v0.8.0-engineering-release-wayland/install_package_and_dependencies.sh
   chmod +x install_package_and_dependencies.sh
   sudo ./install_package_and_dependencies.sh
   ```

5. （可选）如需使用 IMX219 CSI 摄像头，请在昉·星光 2 Debian上在单独开启一个终端窗口，进行 media-pipeline 构建与运行 ISP 控制进程：

   ```
   export PATH=$PATH:/opt/
   /opt/media-ctl-pipeline.sh -d /dev/media0 -i csiphy0 -s ISP0 -a start
   /opt/ISP/stf_isp_ctrl -m imx219mipi -j 0 -a 1
   ```

3. 执行演示代码

通用物体识别基于以下三种模型进行:

3.1. 基于 YOLO-V3 模型的通用物体识别

执行以下步骤，进行基于 YOLO-V3 模型的通用物体识别:

步骤

运行 C ++ 应用:

```
example_dnn_object_detection --config=/usr/share/opencv4/yolo-v3/yolov3-tiny.cfg \
        --model=/usr/share/opencv4/yolo-v3/yolov3-tiny.weights \
        --classes=/usr/share/opencv4/yolo-v3/classes.txt \
--width=416 --height=416 --scale=0.00392 --rgb --target=1 --device=4
```

在 user 用户的任意路径执行以下操作，以在昉·星光 2的 Debian 系统上运行 YOLO-V3 模型的通用物体识别演示代码:

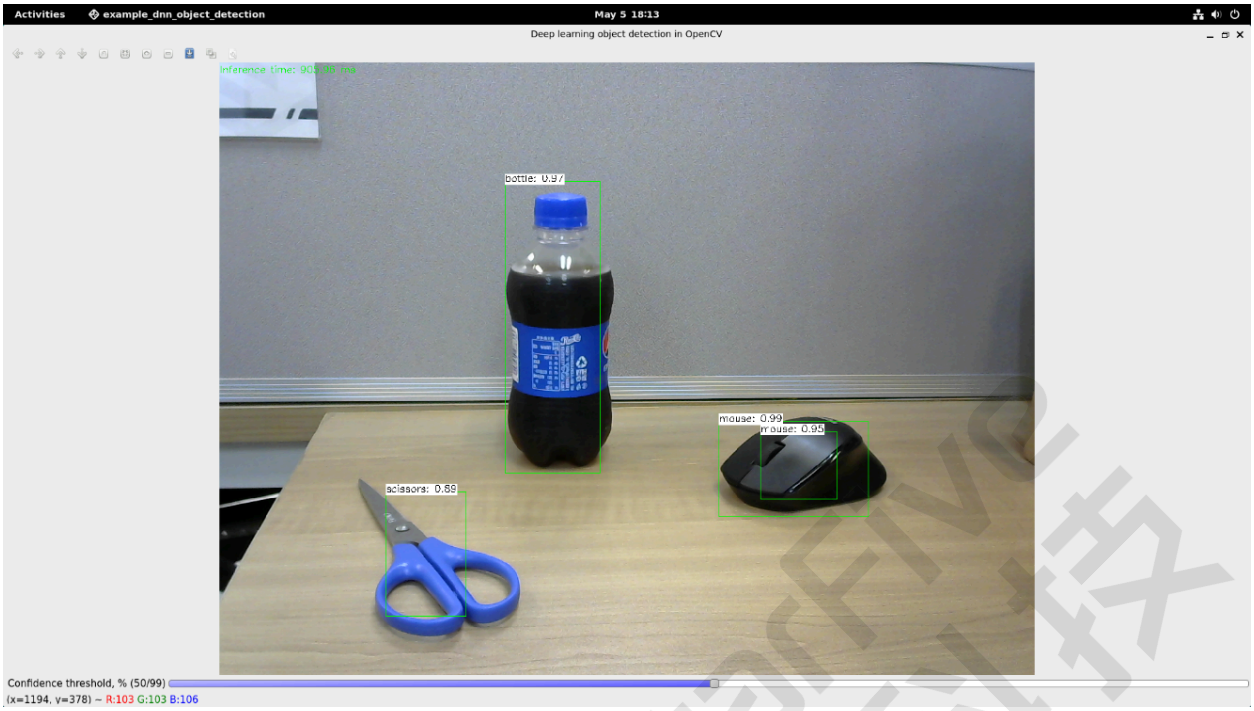
提示:
参数说明:

- 昉·星光 2在原 OpenCV `object_detection.cpp` 应用上新增参数 **device** 用于设置抓取视频流的 video 设备号。一般情况下， `/dev/video1` 为 MIPI CSI 连接的摄像头， `/dev/video4` 则为 USB 摄像头。
- `target = 1` 表示通过 OpenCL 使用GPU加速识别，相较于 target =0 的CPU 运算能大幅提升识别速率，强烈建议保持该选项配置。
- 其余参数可通过`example_dnn_object_detection --help` 查看。

结果

- HDMI 显示器会显示源自摄像头的实时视频流;
- 实时绘制方框，标定模型识别到的物体，并展示其名称与置信度;
- 左上角显示单帧推理时间，换算到推理帧率，约为 1 - 1.5 fps;

图 3-1 基于 YOLO-V3 模型的通用物体识别



3.2. 基于 YOLO-V5 ONNX 模型的通用物体识别

执行以下步骤，进行基于 OLO-V5 ONNX 模型的通用物体识别:

步骤

运行 Python 语言应用:

进入 yolo-v5 第三方 `demo` 目录，以在昉·星光 2的 Debian 系统上运行基于 yolo-5 onnx 模型 Python 演示代码:

```
cd /usr/share/opencv4/yolo-v5/
python3 yolov5.py --device 4
```

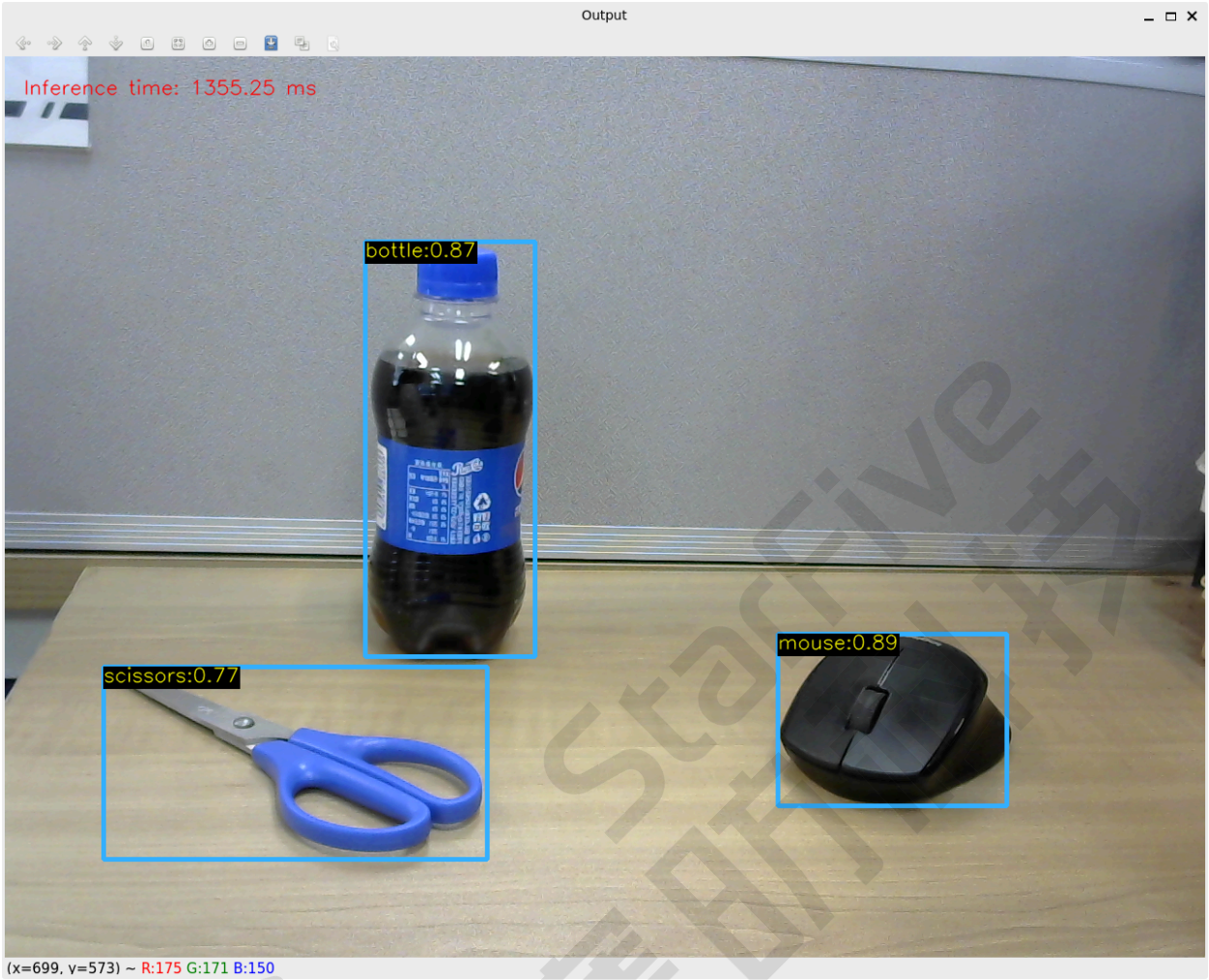
提示:
参数说明:

添加参数 **--device** 加数字 1 或 4 等作为输入参数，指定抓取视频流的 video 设备号。

结果

- HDMI 显示器会显示源自摄像头的实时视频流;
- 实时绘制方框，标定模型识别到的物体，并展示其名称与置信度;
- 左上角显示单帧推理时间，换算到推理帧率，约为 0.6-1.1 fps。

图 3-2 基于 YOLO-V5 ONNX 模型的通用物体识别



3.3. 基于 MobileNet-SSD 模型的通用物体识别

执行以下步骤，进行基于 MobileNet-SSD 模型的通用物体识别：

步骤

运行 C++ 应用：

在 user 用户的任意路径执行以下操作，以在昉·星光 2的 Debian 系统上运行 MobileNet-SSD 模型的通用物体识别演示代码：

```
example_3rd_dnn_mobilenet --device=4
```

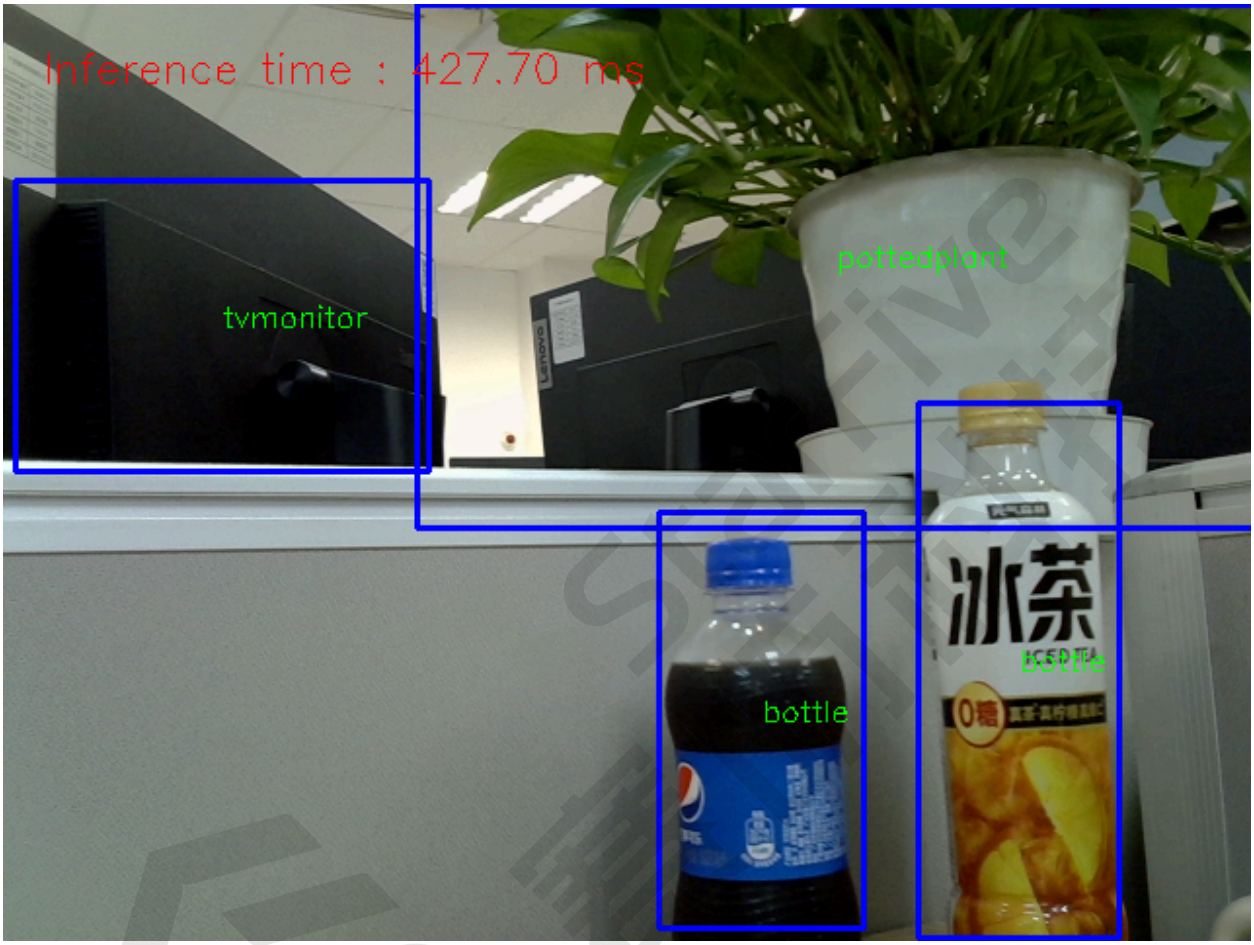
提示：
参数说明：

- 参数 **device** 用于设置抓取视频流的 video 设备号。一般情况下， `/dev/video1` 为 MIPI CSI 连接的摄像头，`/dev/video4` 则为 USB 摄像头。
- 其余参数例如模型路径与阈值设置等，均设定在源码中，尚未提供输入参数修改。

结果

- HDMI 显示器会显示源自摄像头的实时视频流；
- 实时绘制方框，标定模型识别到的物体，并展示其名称；
- 左上角显示单帧推理时间，换算到推理帧率，约为 2.5-3fps；

图 3-3 基于 MobileNet-SSD模型的通用物体识别



4. 演示源代码

本演示中的资源代码仅作为参考。

开发者在安装昉·星光 2 Debian 物体识别软件安装包后，本项目所涉及的动态库、头文件、应用源代码均会被安装到昉·星光 2 Debian 系统中，开发者可按需进行二次开发，以探索更多的可能：

- OpenCV 动态库路径：

```
/usr/lib/riscv64-linux-gnu/
```

- OpenCV 头文件路径：

```
/usr/include/opencv4/
```

- OpenCV 官方物体识别 demo 源码路径，本文用于 yolo-v3 模型的应用：

```
/usr/share/doc/opencv-doc/examples/dnn/object_detection.cpp
```

- 涉及的第三方应用来源：
  - YOLO-V5
  - MobileNet-SSD

5. 目标识别系列应用

赛昉科技实现的目标识别系列应用所使用的视觉框架基于OpenCV，赛昉科技对其进行了昉·星光 2的平台适配与底层GPU加速调优。

该系列应用包括:

- 使用昉·星光 2进行通用物体识别
- 使用昉·星光 2进行二维码检测与解码
- 使用昉·星光 2检测图像边缘
- 使用昉·星光 2检测图像缺陷

6. 资源下载

点击本栏找到所有的代码下载资源。

本页包括所有赛昉科技提供的代码下载资源。

- RVspace Wiki
- 应用中心
- 文档中心
- 技术论坛
- 昉·星光 2 GitHub代码仓
- 昉·星光 2 Debian操作系统下载
- 代码下载（赛昉科技官方GitHub页面）
- 所有开源技术文档

7. 立即购买

点击本栏获取在线购买链接和配件购买链接。

购买单板计算机

点击以下页面，您可以找到所在地区的经销商，或覆盖全球的销售渠道，以购买昉·星光 2单板计算机。

- 购买昉·星光2开发板

购买配件

点击以下页面，您可以找到所有昉·星光 2单板计算机已验证适配的配件及其购买链接。

- 购买配件