



**StarFive**  
**赛昉科技**

# **Using VisionFive 2 to Recognize General Objects**

Application Notes

Version: 1.0

Date: 2023/06/30

Doc ID: VisionFive 2-ANEN-07

# Legal Statements

Important legal notice before reading this documentation.

## PROPRIETARY NOTICE

Copyright©Shanghai StarFive Technology Co., Ltd., 2023. All rights reserved.

Information in this document is provided "as is," with all faults. Contents may be periodically updated or revised due to the product development. Shanghai StarFive Technology Co., Ltd.(hereinafter "StarFive") reserves the right to make changes without further notice to any products herein.

StarFive expressly disclaims all warranties, representations, and conditions of any kind, whether express or implied, including, but not limited to, the implied warranties or conditions of merchantability, fitness for a particular purpose and non-infringement.

StarFive does not assume any liability rising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation indirect, incidental, special, exemplary, or consequential damages.

All material appearing in this document is protected by copyright and is the property of StarFive. You may use this document or any part of the document for internal or educational purposes only, provided you do not modify, edit or take out of context the information in this document in any manner. Information contained in this document may be used, at your sole risk, for any purposes. StarFive authorizes you to copy this document, provided that you retain all copyright and other proprietary notices contained in the original materials on any copies of the materials and that you comply strictly with these terms. This copyright permission does not constitute an endorsement of the products or services.

## Contact Us

Address: Room 502, Building 2, No. 61 Shengxia Rd., China (Shanghai) Pilot Free Trade Zone, Shanghai, 201203, China Room 502, Building 2, No. 61 Shengxia Rd., China (Shanghai) Pilot Free Trade Zone, Shanghai, 201203, China

Website: <http://www.starfivetech.com> <http://www.starfivetech.com>

Email: [sales@starfivetech.com](mailto:sales@starfivetech.com)(sales) , [support@starfivetech.com](mailto:support@starfivetech.com)(support)



---

# Contents

List of Tables.....	4
List of Figures.....	5
Legal Statements.....	ii
Preface.....	vi
<b>1. Introduction.....</b>	<b>7</b>
<b>2. Preparation.....</b>	<b>8</b>
2.1. Environment Requirements.....	8
2.2. Preparing Hardware.....	8
2.2.1. Hardware Setup.....	8
2.3. Preparing Software.....	9
<b>3. Running Demo Codes.....</b>	<b>11</b>
3.1. General Object Recognition based on YOLO-V3 Model.....	11
3.2. General Object Recognition based on YOLO-V5 Model.....	12
3.3. General Object Recognition based on MobileNet-SSD Model.....	13
<b>4. Demo Source Code.....</b>	<b>15</b>
<b>5. Object Detection Applications.....</b>	<b>16</b>
<b>6. Code Download.....</b>	<b>17</b>
<b>7. Buy Now.....</b>	<b>18</b>

## List of Tables

Table 0-1 Version History.....	vi
Table 2-1 Hardware Preparation.....	8



## List of Figures

Figure 2-1 VisionFive 2 Block Diagram - Top View.....	9
Figure 3-1 General Object Recognition based on YOLO-V3 Model.....	12
Figure 3-2 General Object Recognition based on YOLO-V5 ONNX Model.....	13
Figure 3-3 General Object Recognition based on MobileNet-SSD Model.....	14



# Preface

About this guide and technical support information.

## About this document

This application notes provide users with steps to use VisionFive 2 to recognize various objects based on OpenCV. StarFive has made it compatible with VisionFive 2 and accelerated and improved GPU performance.

This application notes include three applications correspond to three object recognition models:

- YOLO-V3
- YOLO-V5 with ONNX format converted from PyTorch
- MobileNet-SSD






## Version History

Table 0-1 Version History

Version	Release Date	Revision
1.0	2023/06/ 30	The first official release.

## Notes and notices

The following notes and notices might appear in this guide:

-  **Tip:**  
Suggests how to apply the information in a topic or step.
-  **Note:**  
Explains a special case or expands on an important point.
-  **Important:**  
Points out critical information concerning a topic or step.
-  **CAUTION:**  
Indicates that an action or step can cause loss of data, security problems, or performance issues.
-  **Warning:**  
Indicates that an action or step can result in physical harm or cause damage to hardware.

---

# 1. Introduction

This application notes provide users with steps to use VisionFive 2 to recognize various objects based on OpenCV. StarFive has made it compatible with VisionFive 2 and accelerated and improved GPU performance.

This application notes include three applications correspond to three object recognition models:

- YOLO-V3
- YOLO-V5 with ONNX format converted from PyTorch
- MobileNet-SSD

This application is one of the object detection applications implemented by StarFive. For the other applications, see [Object Detection Applications \(on page 16\)](#).



## 2. Preparation

Before executing the demo program, make sure you prepare the following:

### 2.1. Environment Requirements

The environment requirements are as follows:

- Linux Kernel: Linux 5.15
- OS: Debian 12
- SBC: VisionFive 2
- SoC: JH7110

### 2.2. Preparing Hardware

Before executing the demo program, make sure you prepare the following:

Table 2-1 Hardware Preparation

Type	M/O*	Item	Notes
General	M	VisionFive 2 Board	-
General	M	<ul style="list-style-type: none"><li>• 32 GB (or more) micro-SD card</li><li>• Micro-SD card reader</li><li>• Computer (Windows/Mac OS/Linux)</li><li>• USB to serial converter (3.3 V I/O)</li><li>• Ethernet cable</li><li>• Power adapter (5 V / 3 A)</li><li>• USB Type-C Cable</li></ul>	These items are used for flashing Debian OS into a Micro-SD card.
Object Detection Applications	M	Monitor and accessories: <ul style="list-style-type: none"><li>• An HDMI monitor</li><li>• An HDMI cable</li></ul>	-
	M	Camera and accessories: <ul style="list-style-type: none"><li>• A USB camera</li><li>• A IMX219 MIPI camera</li></ul>	For the compatible USB cameras, refer to <a href="#">JH7110 AVL</a> .
	M	A USB keyboard	Used to operate on the Debian terminal.
	O	A USB mouse	Used to operate on the Debian terminal.



**Note:**

\*: M: Mandatory, O: Optional

#### 2.2.1. Hardware Setup

Connect the external devices and accessories and power adapter to VisionFive 2 according to the following top-view diagram:



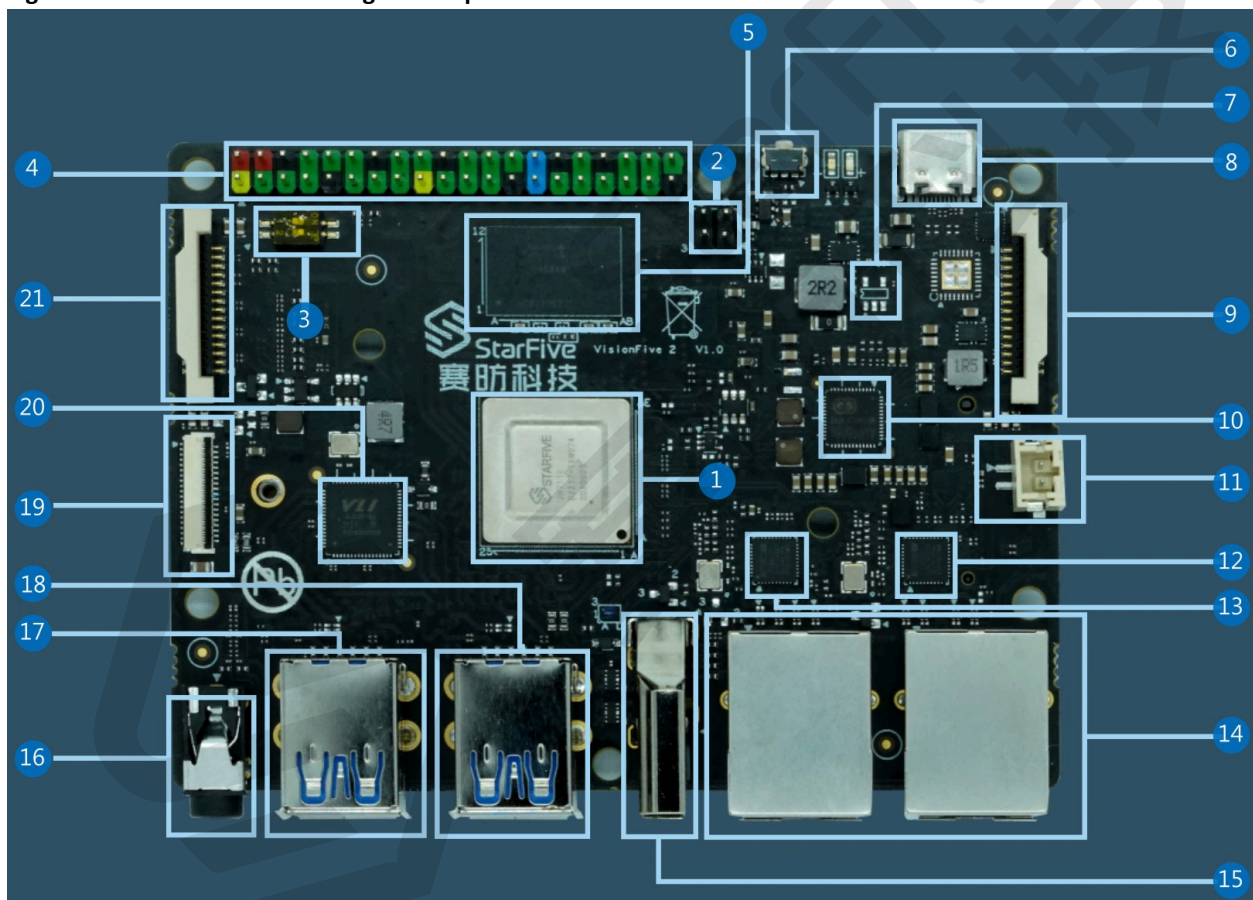
1. Plug the Micro-SD card with the latest Debian image into the card slot on the back of VisionFive 2.

**Note:**

Flash Debian OS into a Micro-SD card as described in the *Flashing Debian OS to a Micro-SD Card* section in the [VisionFive 2 Single Board Computer Quick Start Guide](#). Make sure that the Debian image you download is the latest version. creation time of the folder where the image is located. Select the image with the latest creation time in the folder.

2. Connect the Ethernet cable to one of the RJ45 Ethernet ports (No. 14).
3. Connect the HDMI monitor with the HDMI cable to the HDMI 2.0 port (No. 15).
4. Connect the IMX219 MIPI camera to HDMI 2.0 port (No. 9). If the camera has a pin header for fixing purposes (without connecting to any actual circuit), it can be connected to the available pins of the 40-pin GPIO header (No. 4). (Optional)
5. Connect the USB camera, keyboard, and mouse to the USB ports (No. 17 and 18).
6. Connect the power adapter with a USB cable to the USB-C port (No. 8). (This step must be performed after Steps 1 to 4).

Figure 2-1 VisionFive 2 Block Diagram - Top View



## 2.3. Preparing Software

Make sure the following procedures are performed:

**Note:**

Make sure that you use the latest Debian image.

## | 2 - Preparation

---

1. Log into the Debian and make sure VisionFive 2 is connected to the Internet. For detailed instructions, refer to the *Using SSH over Ethernet* or *Using a USB to Serial Converter* section in the [VisionFive 2 Single Board Computer Quick Start Guide](#).
2. Log into the Debian and make sure VisionFive 2 is connected to the Internet. Log into Debian with keyboard, mouse and monitor connected. For detailed instructions, refer to *Using Desktop over HDMI* section in the [VisionFive 2 Single Board Computer Quick Start Guide](#).
3. Extend the partition on Debian as described in *Extend Partition* section in the [VisionFive 2 Single Board Computer Quick Start Guide](#).
4. Execute the following command on VisionFive 2 Debian to install StarFive Packages and its dependencies. The installation requires 1-3 hours.

```
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. (Optional) If you need to use the IMX219 CSI camera, open a separate terminal window on VisionFive 2 Debian and build media pipeline and run the ISP control process:

```
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. Running Demo Codes

The general object recognition is based on the following three models:

- [General Object Recognition based on YOLO-V3 Model \(on page 11\)](#)
- [General Object Recognition based on YOLO-V5 Model \(on page 12\)](#)
- [General Object Recognition based on MobileNet-SSD Model \(on page 13\)](#)

### 3.1. General Object Recognition based on YOLO-V3 Model

Execute the following steps to recognize general objects based on YOLO-V3 Model:

#### Step:

To run C++ application:

```
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
```

Perform the following step under the `user` path on VisionFive 2 Debian to run YOLO-V3 general object recognition demo:



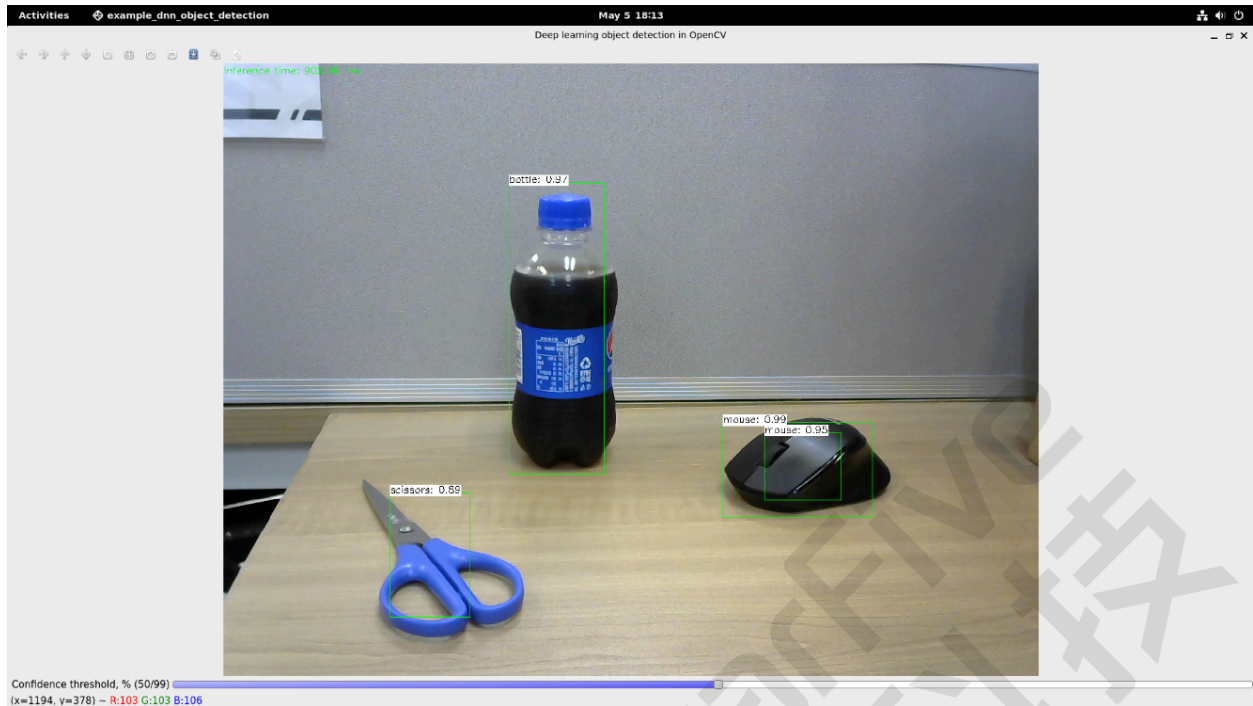
#### Tip:

Parameter Information:

- Parameter `device` is added to the OpenCV `object_detection.cpp` application to set the video device number for capturing a video stream. Normally, `/dev/video1` represents the camera connected to MIPI CSI, and `/dev/video4` represents the USB camera.
- `target=1` represents that OpenCL uses GPU to accelerate the recognition, which will greatly increase the speed compared with using `target=0`. Strongly recommends that you keep this parameter in the command.
- Other parameters can be checked by executing `example_dnn_object_detection --help`.

#### Result:

- HDMI monitor displays the real-time video stream from a webcam;
- HDMI monitor displays real-time drawings of bounding boxes, labeling the objects recognized by the model, and displaying their names and confidence level;
- The upper left corner displays the inference time per frame, which translates to an approximate inference frame rate of 1 - 1.5 fps;

**Figure 3-1 General Object Recognition based on YOLO-V3 Model**

### 3.2. General Object Recognition based on YOLO-V5 Model

Execute the following steps to recognize general objects based on YOLO-V5 Model:

#### Step:

To run Python application:

Enter the third-party application directory, `demo`, of YOLO-V5 and execute the command to run the Python application:

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



#### Tip:

Parameter Information:

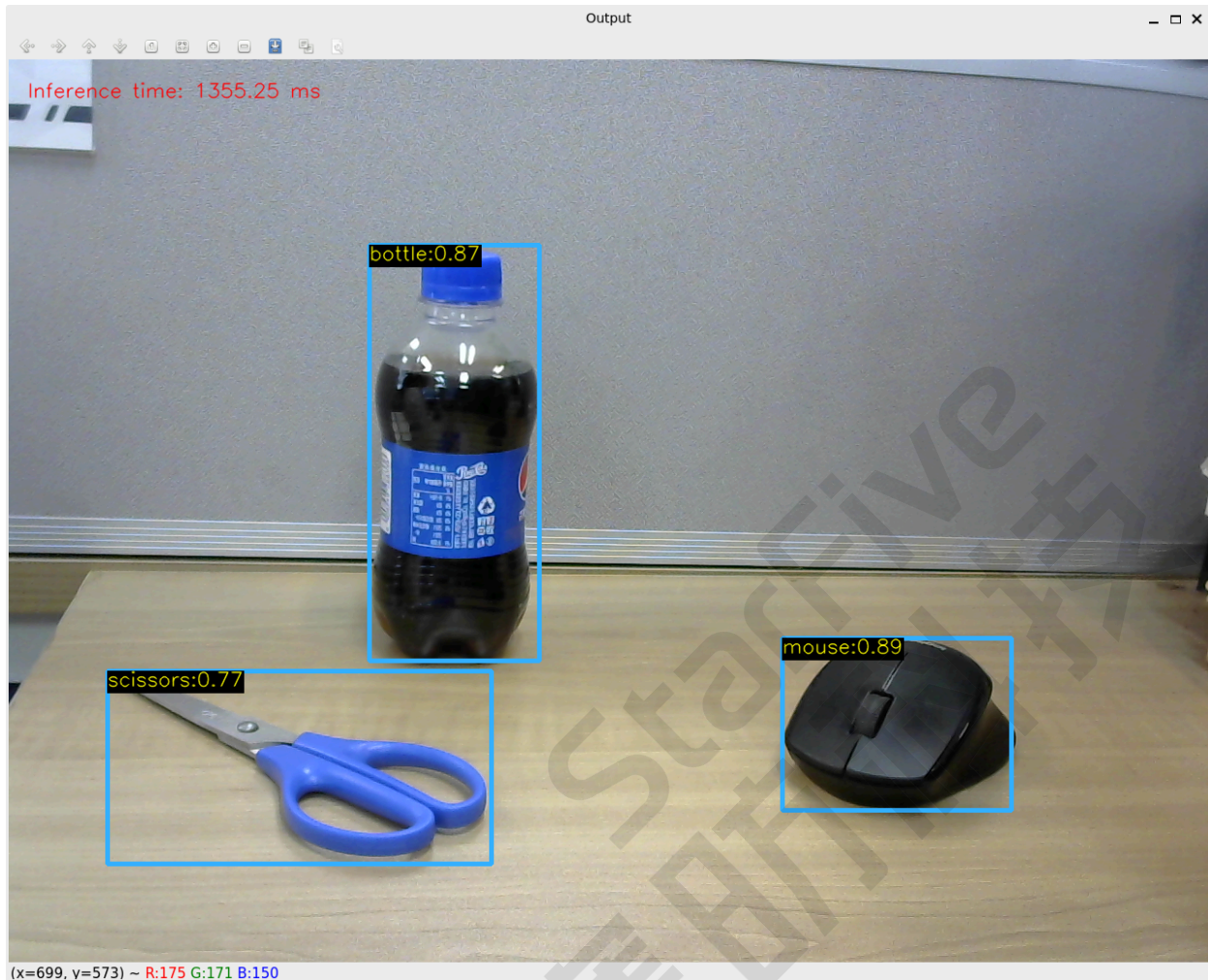
Added `--device` and number 1 or 4 as an input parameter to specify the number of the video device capturing a video stream.

#### Result:

- HDMI monitor displays the real-time video stream from a webcam;
- HDMI monitor displays real-time drawings of bounding boxes, labeling the objects recognized by the model, and displaying their names and confidence level;
- The inference time per frame is displayed on the upper left corner, which translates to an approximate inference frame rate of 0.6 to 1.1 fps;



Figure 3-2 General Object Recognition based on YOLO-V5 ONNX Model



### 3.3. General Object Recognition based on MobileNet-SSD Model

Execute the following steps to recognize general objects based on MobileNet-SSD Model:

#### Step:

To run C++ application:

To run the demo code for general object recognition based on MobileNet-SSD on VisionFive 2 Debian, execute the following command under any path under user:

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



#### Tip:

Parameter Information:

- The parameter **device** is to set the video device number for capturing the video stream. Normally, `/dev/video1` represents the camera connected to MIPI CSI, and `/dev/video4` represents the USB camera.
- Other parameters, such as mode path and thread value settings, are set in the source code, and configuration is not allowed.

**Result:**

- HDMI monitor displays the real-time video stream from a webcam;
- HDMI monitor displays real-time drawings of bounding boxes, labeling the objects recognized by the model, and displaying their names;
- The inference time per frame is displayed on the upper left corner, which translates to an approximate inference frame rate of 2.5-3 fps;

**Figure 3-3 General Object Recognition based on MobileNet-SSD Model**



---

## 4. Demo Source Code

The Python source code of this demo is provided for reference purposes only.

After installing VisionFive 2 Debian object detection package, the dynamic library, header files, and application source code are also installed in the VisionFive 2 Debian. Developers can engage in secondary development as needed to unlock more possibilities.

- OpenCV dynamic library path:

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

- OpenCV head file path:

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

- The path of OpenCV object recognition demo source code, which is used for YOLO-V3 application in the document:

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

- The third party application sources:

- [YOLO-V5](#)
- [MobileNet-SSD](#)

---

## 5. Object Detection Applications

These applications use a vision framework based on OpenCV. StarFive has made it compatible with VisionFive 2 and accelerated and improved GPU performance.

These applications include:

- [Using VisionFive 2 to Recognize General Objects](#)
- [Using VisionFive 2 to Detect and Decode QR Codes](#)
- [Using VisionFive 2 to Detect Image Edge](#)
- [Using VisionFive 2 to Detect Image Defects](#)





---

## 6. Code Download

Click on this tab to find all code download resources.

The following link contains all the code download resources from StarFive.

- [StarFive Official GitHub Page](#)



---

## 7. Buy Now

Click on this tab to find all the online shops and compatible accessories.

### Buy SBC

Use the following page to find your nearest sales channel or the global channels for purchasing a VisionFive 2 Single Board Computer (SBC).

- [Buy VisionFive 2](#)

### Buy Parts

Use the following page to find the parts that are tested as compatible with VisionFive 2.

- [Buy Accessory](#)

