

Avalue Technology Co., Ltd

AIB-NINX AI Model

Python Sample v1.0

February 18, 2024

# Directory

<b>1</b>	<b><i>Introduction to Hardware Architecture.....</i></b>	<b><i>1</i></b>
1.1	Hardware.....	3
<b>2</b>	<b><i>AI Model DescriptionAI .....</i></b>	<b><i>3</i></b>
2.1	Humanoid Detection.....	3
2.2	Vehicle Detection .....	5
2.3	Face Detection.....	6
2.4	License Plate Detection .....	8
2.5	Humanoid Detection and Skeletal Diagrams.....	10
<b>3</b>	<b><i>General parameter setting.....</i></b>	<b><i>11</i></b>
3.1	RESULT OF FUNCTION.....	12
3.2	OBJECT GPU TYPE.....	12
3.3	OBJECT DETECT CONFIG MODEL .....	12
3.4	Confirm the AI Model Build.....	14
<b>4</b>	<b><i>Precautions and Exception Elimination .....</i></b>	<b><i>15</i></b>
4.1	Ubuntu .....	15
4.2	Ubuntn Suite.....	15
4.3	Exception Error Handling .....	16
<b>5</b>	<b><i>GITHUB FILE DOWNLOAD AND DECOMPRESSION .....</i></b>	<b><i>19</i></b>
5.1	GitHub file download (example) .....	20

# 1 INTRODUCTION TO HARDWARE ARCHITECTURE

The AIR6N0 system uses NVIDIA's Jetson Orin™ series, incorporating Jetson Orin™ NX embedded modules, and provides 100 TOPS of artificial intelligence image processing computing power, product appearance and specifications as shown below. °



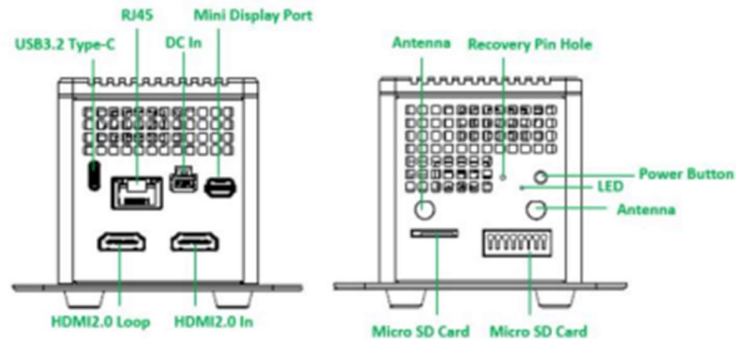
System				
SOM	NVIDIA Jetson Orin™ NX 16GB	NVIDIA Jetson Orin™ NX 8GB	NVIDIA Jetson Orin™ Nano 8GB	NVIDIA Jetson Orin™ Nano 4GB
CPU	8-Core Arm® Cortex®-A78AE v8.2 64-Bit CPU 2MB L2 + 4MB L3	6-Core Arm® Cortex®-A78AE v8.2 64-Bit CPU 1.5MB L2 + 4MB L3	6-Core Arm® Cortex®-A78AE v8.2 64-Bit CPU 1.5MB L2 + 4MB L3	
GPU	1024-Core NVIDIA Ampere Architecture GPU with 32 Tensor Cores		1024-core NVIDIA Ampere Architecture GPU with 32 Tensor Cores	512-core NVIDIA Ampere Architecture GPU with 16 Tensor Cores
AI Performance	100 TOPS	70 TOPS	40 TOPS	20 TOPS
System Memory	16GB 128-bit LPDDRS 102.4GB/s	8GB 128-bit LPDDRS 102.4GB/s	8GB 128-bit LPDDRS 68 GB/s	4GB 64-bit LPDDRS 34 GB/s
Interface				
Storage	Supports External NVMe 1×Micro SD Card Slot			
Display Interface	1×Mini DP1.4			
Ethernet	1×RJ45 for 10/100/1000Mbps Ethernet (HDCP Client)			
Expansion Slot	1×M.2 2230 M Key PCIe Gen4×2 Slot (Nano: Gen3×2 ) 1×M.2 2230 E Key PCIe Gen4×1 Slot (Nano: Gen3×1 )			
USB	1×USB3.2 Gen2 ( Type-C )			
MIPI	8 MIPI CSI-2 Lanes ( D-PHY 2.1, Support MIPI Camera, Capture Card )			
Peripheral Communication	8 Pin Phoenix Connector 1×GPIO 1×RS232 1×I2C			
Misc. Features	Firmware Upgradable			
Development Environment				
OS	Ubuntu: 20.04			
Kernel	5.10.104-tegra or Higher			
BSP	Linux for Tegra(L4T) R35.3.1 or Higher			
SDK	JetPack 5.1.1 or Higher			
Environment				
Power Supply	DC input : 9~36V			
Power Consumption	MAX 30W (With SOM)			
Operating Temperature	0~60 °C with Airflow			
Storage Temperature	-20~80 °C			
Certification	CE, FCC			

## AIB-NINX AI Model Python Sample

### SDK / Software

Video Feature				
SOM	NVIDIA Jetson Orin™ NX 16GB	NVIDIA Jetson Orin™ NX 8GB	NVIDIA Jetson Orin™ Nano 8GB	NVIDIA Jetson Orin™ Nano 4GB
Video Encode	AV1 ( UHP ) 1x4K60   3x4K30   6x1080p60   12x1080p30	1080p30 supported by 1-2 CPU cores		
	H.265 ( UHP ) 1x4K60   3x4K30   6x1080p60   12x1080p30			
	H.264 ( UHP ) 1x4K60   2x4K30   5x1080p60   11x1080p30			
Video Decode	AV1 ( Main Profile ) 1x8K30   2x4K60   4x4K30   10x1080p60   20x1080p30	AV1 ( Main Profile ) 1x4K60   2x4K30   5x1080p60   10x1080p30		
	H.265 ( Main, Main10 ) 1x8K30   2x4K60   4x4K30   9x1080p60   18x1080p30	H.265 ( Main, Main10 ) 1x4K60   2x4K30   5x1080p60   11x1080p30		
	H.264 ( Baseline, Main, High ) 1x4K60   2x4K30   5x1080p60   11x1080p30	H.264 ( Baseline, Main, High ) 1x4K30   3x1080p60   7x1080p30		
	VP9 ( Profile 0, Profile 2 ) 1x4K60   3x4K30   7x1080p60   15x1080p30	VP9 ( Profile 0, Profile 2 ) 1x4K60   2x4K30   5x1080p60   11x1080p30		
SDK				
QCAP	Capture High Performance Renderer Image Snapshot De-interlace, Alpha Blending Engine Auto Signal Detection 2D/3D Video, Audio and VANC Streams Capture			
	Record Encrypt / Sync / Clone / Recording Time-Shifting / Rewind / Pre-Event / Recording Multi-Streams ( 3D ) Recording Animation Transition Effect Video Cropping, Scaling and Alpha Blending Engine			
	Stream 2D/3D Universal Stream Client 2D/3D Multi-Streams Stream Server RTSP, RTMP, HLS, SRT, TS, WebRTC. NDI-HX (*), Full NDI (*), Dante AV-H (*) Animation Transition Effect Video Cropping, Scaling and Alpha Blending Engine *Separate License Required			
QDEEP	AI SDK Integrated Multiple Algorithms and Deep-Learning Models in Various Fields of Applications Face Recognition Objects Detection Objects Segment Optical Character Recognition License Plate Recognition Customizable Video AI Functions Upon Request			
Software (Optional)				
Xstreamer	Web Based User Interface			
	Encode / Decode AV1, H.26X			
	Color Format Adjust 444 / 422 / 420, 10Bit / 8Bit Select			
	Record MP4, TS			
	Stream / Network RTSP, RTMP, HLS, SRT, TS, WebRTC. NDI-HX (*), Full NDI (*), Dante AV-H (*) *Separate License Required			

## 1.1 Hardware



Notes:

LED (RGB): indicates that the device is powered off and standby

LED (green): Indicates that the device starts up normally

## 2 AI MODEL DESCRIPTIONAI

Using a variety of AI models provided by AIB-NINX, real-time image capture and AI recognition technology are used to achieve situational simulation and AI analysis demonstration.

The project is developed in C++ and Python to improve development flexibility and facilitate subsequent developers to apply it to different identification scenarios. °

### 2.1 Humanoid Detection

Humanoid detection mainly recognizes humanoid objects in the real-time video through the external Webcam, then marks and selects them, records the movement trajectory, and can also be identified by loading videos.

Application/Applicable Scenarios :

- ✚ Intelligent monitoring and security systems
- ✚ Traffic monitoring and pedestrian detection
- ✚ Crowd analysis for events and exhibitions

### 1. Example file name :

RELEASES.QDEEP.MODEL.TINY.PERSON 1.1.0.202.4.7z

### 2. Unzip the instruction :

7z e RELEASES.QDEEP.MODEL.TINY.PERSON\ 1.1.0.202.4.7z -omodel/tiny\_person

### 3. Create a folder name :

tiny\_person

### 4. Python Template Adjustment :

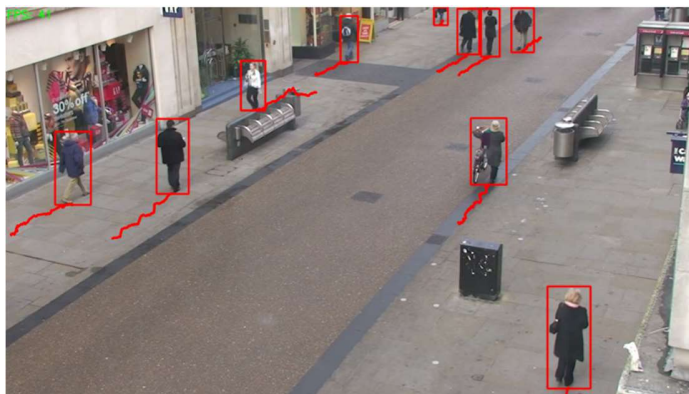
#### ○ Module Path :

```
strModelName = './model/tiny_person/QDEEP.OD.TINY.PERSON.CFG'
```

#### ○ AI Module :

```
QDEEP.QDEEP_CREATE_OBJECT_DETECT(QDeepGPType.DEFAULT,0,QDeepObject  
DetectConfigModel.QDEEP_OBJECT_DETECT_CONFIG_MODEL_CUSTOMIZED_LITE,  
strModelName.encode('utf-8'),byref(self.m_detector),QDeepObjectDetectFlag.  
TRAJECTORY_TRACKING)
```

#### ○ Display View :



### 5. Suggestions for decompression :

- It is recommended to install Webcam first and confirm that the driver is installed successfully.




### 6. GitHub download link :

[Avalue-Technology/edge.ai.jetson.orin.human.detection>About Jetson Orin AI Model - Human Detection Sample.](#)

## 2.2 Vehicle Detection

Vehicle detection is mainly used by external Webcam to identify the moving vehicle in the real-time video, then mark and select the box, record the movement trajectory, and can also be identified by loading the video.

Application/Applicable Scenarios :

-  Intelligent monitoring and security systems
-  Smart City & Transportation Data Analytics
-  Vehicle security and anti-theft tracking

### 1. Example file name :

RELEASES.QDEEP.MODEL.TAIWAN.TRAFFIC.C4.TINY 1.1.0.202.4.7z

### 2. Unzip the instruction :

```
7z e RELEASES.QDEEP.MODEL.TAIWAN.TRAFFIC.C4.TINY\1.1.0.202.4.7z -omodel/  
taiwan_traffic_C4
```

### 3. Create a folder name :

taiwan\_traffic\_C4

### 4. Python Template Adjustment :

- **Module Path :**

```
strModelName='./model/taiwan_traffic_C4/QDEEP.OD.TAIWAN.TRAFFIC.C4.TINY.CF  
G'
```

- **AI Module :**

```
QDEEP.QDEEP_CREATE_OBJECT_DETECT(QDeepGPUPType.NVIDIA,0,  
QDeepObjectDetectConfigModel.QDEEP_OBJECT_DETECT_CONFIG_MODEL_TRAFFI  
C_TAIWAN_04, strModelName.encode('utf-8'), byref(self.m_detector), 0)
```

- **Display View :**



### 5. Suggestions for decompression :

- It is recommended to install Webcam first and confirm that the driver is installed successfully.

### 6. GitHub download link :




[Avalue-Technology/edge.ai.jetson.orin.vehicle.detection>About Jetson Orin AI Model - Vehicle Detection Sample.](#)

## 2.3 Face Detection

Face detection is mainly used by external Webcam to identify the humanoid face area in the real-time image, and then mark and select the frame, or it can be identified by loading videos, and then it can be used through facial feature recognition.



Application/Applicable Scenarios :

-  Authentication and access management
-  Public Safety & Surveillance
-  Smart medical care and health monitoring

1. **Example file name :**

RELEASES.QDEEP.MODEL.FACE.LANDMARK.5KPS 1.1.0.203.1.7z

2. **Unzip the instruction :**

```
7z e RELEASES.QDEEP.MODEL.FACE.LANDMARK.5KPS\1.1.0.202.4.7z -omodel/  
FACE_LANDMARK_5KPS
```

3. **Create a folder name :**

FACE\_LANDMARK\_5KPS

4. **Python Template Adjustment :**

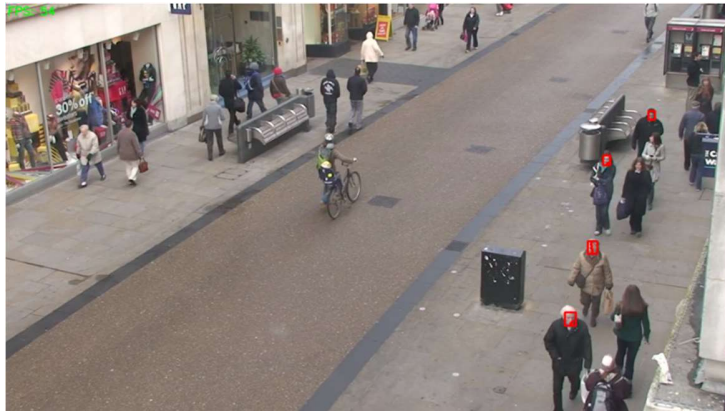
- **Module Path :**

```
strModelName='./model/FACE_LANDMARK_5KPS/QDEEP.OD.FACE.LANDMARK.5K  
PS.CFG'
```

- **AI Module:**

```
QDEEP.QDEEP_CREATE_OBJECT_DETECT(QDeepGPUType.NVIDIA,  
0,QDeepObjectDetectConfigModel.QDEEP_OBJECT_DETECT_CONFIG_MODEL_FACE  
_LANDMARK_5_KEYPOINTS, strModelName.encode('utf-8'),byref(self.m_detector),  
QDeepObjectDetectFlag.FULL | QDeepObjectDetectFlag.BEHAVIOR))
```

- **Display View :**



### 5. Suggestions for decompression :

- It is recommended to install Webcam first and confirm that the driver is installed successfully.

### 6. GitHub download link :

[Avalue-Technology/edge.ai.jetson.orin.face.detection: About Jetson Orin AI Model - Facial Detection Sample.](#)

## 2.4 License Plate Detection

License plate detection mainly uses external Webcam to search for the location of the license plate for the vehicle in the real-time video, and then marks and selects the box, or can be identified by loading the video, and then the relevant vehicle number can be confirmed through the license plate recognition.

Application/Applicable Scenarios :

- 🚦 Intelligent traffic management
- 🚦 Smart Parking Management
- 🚦 Enterprise & Community Access Control Management

### 1. Example file name :

RELEASES.QDEEP.MODEL.LICENSE.PLATE.RECOGNITION.LAW.TINY 1.1.0.202.4.7z

### 2. Unzip the instruction :

```
7z e RELEASES.QDEEP.MODEL.LICENSE.PLATE.RECOGNITION.LAW.TINY\ 1.1.0.202.4.7z -  
omodel/LPR_LAW_ENFORCEMENT
```

### 3. Create a folder name :

LPR\_LAW\_ENFORCEMENT

### 4. Python Template Adjustment :

- Module Path :

```
strModelName='./model/LPR_LAW_ENFORCEMENT/QDEEP.OD.LICENSE.PLATE.REC  
OGNITION.LAW.TINY.CFG'
```

- AI Module :

```
QDEEP.QDEEP_CREATE_OBJECT_DETECT(QDeepGPUPType.NVIDIA,0,  
QDeepObjectDetectConfigModel.QDEEP_OBJECT_DETECT_CONFIG_MODEL_LICENS  
E_PLATE_RECOGNITION_LAW_ENFORCEMENT,strModelName.encode('utf-8'),  
byref(self.m_detector),QDeepObjectDetectFlag.TRAJECTORY_TRACKING|  
QDeepObjectDetectFlag.FEATURE_VECTOR)
```

- Display View :



### 5. Suggestions for decompression :

- It is recommended to install Webcam first and confirm that the driver is installed successfully.




### 6. GitHub download link :

[Avalue-Technology/edge.ai.jetson.orin.anpr>About JetsonOrinAI Model - Automatic number-plate recognition Sample.](#)

## 2.5 Humanoid Detection and Skeletal Diagrams

Humanoid detection and skeletal mapping are mainly used by external Webcam to detect and draw human skeleton icons in real-time images, which can be adjusted according to humanoid movements, and can also be identified by loading videos.

Application/Applicable Scenarios :

-  Sports & Exercise Analysis
-  Safety monitoring and behavior analysis
-  Human-computer interaction and intelligent control

### 1. Example file name :

RELEASES.QDEEP.MODEL.HUMAN.SKELETON.17KPS 1.1.0.203.1.7z

### 2. Unzip the instruction :

7z e RELEASES.QDEEP.MODEL.HUMAN.SKELETON.17KPS\ 1.1.0.203.1.7z –  
omodel/LPR\_LAW\_ENFORCEMENT

### 3. Create a folder name :

Human\_Skeleton\_17\_Keypoin

### 4. Python Template Adjustment :

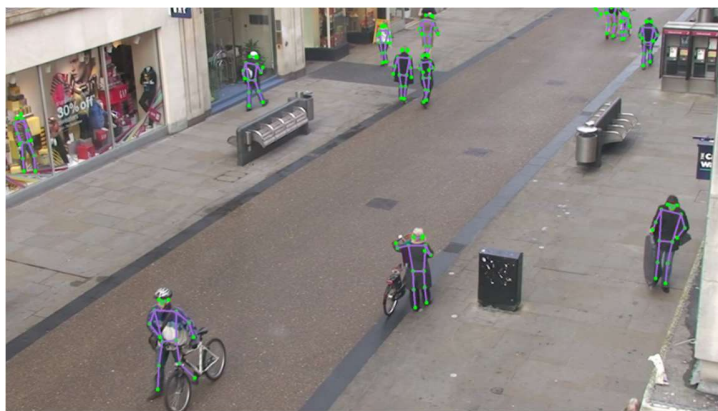
- Module Path :

```
self.model_path='./model/Human_Skeleton_17_Keypoints/QDEEP.OD.HUMAN.SKELETON.17KPS.CFG'
```

- **AI Module :**

```
QDEEP_PERSON.QDEEP_CREATE_OBJECT_DETECT(0x00000001,0,  
QDEEP_OBJECT_DETECT_CONFIG_MODEL_CUSTOMIZED_LITE,  
self.model_person_path.encode('utf-8'),byref(self.m_person_detector),  
QDEEP_OBJECT_DETECT_FLAG_TRAJECTORY_TRACKING, None)
```

- **Display View :**



### 5. Suggestions for decompression :

- It is recommended to install Webcam first and confirm that the driver is installed successfully.

### 6. GitHub download link :

[Avalue-Technology/edge.ai.jetson.orin.human.keypoint.detection:  
About Jetson Orin AI Model - Human Keypoint Detection Sample.](#)

## 3 GENERAL PARAMETER SETTING

Python's enum module defines the corresponding enumeration categories, adjusts and modifies the C++ enumeration parameters in the example.

### 3.1 RESULT OF FUNCTION

```
class QDeepObjectDetectFlag(IntFlag):
    TRAJECTORY_TRACKING = 0x00000001
    SUB_CLASS           = 0x00000002 # ONLY USED: FACE LANDMARK (AGE & GENDER)
    FEATURE_VECTOR      = 0x00000004
    BEHAVIOR            = 0x00000008 # ONLY USED: FACE LANDMARK (EMOTION)
    EXTRA_ATTRIBUTE    = 0x00000010 # ONLY USED: FACE LANDMARK #5 (ROLL / YAW)
                                # ONLY USED: FACE LANDMARK #68 (ROLL / YAW / PITCH)
    FULL                = TRAJECTORY_TRACKING | SUB_CLASS | FEATURE_VECTOR # 0x00000007
```

### 3.2 OBJECT GPU TYPE

```
class QDeepGPUType(IntFlag):
    DEFAULT      = 0x00000001
    NVIDIA       = 0x00000001
    INTEL_CPU    = 0x00000002
    INTEL_GPU    = 0x00000004
    INTEL_VPU_MOVIDIUS = 0x00000008 # MYRAID X / MYRAID 2 / NCS
```

### 3.3 OBJECT DETECT CONFIG MODEL

```
class QDeepObjectDetectConfigModel(IntEnum):
    # |NVIDIA|INTEL|CLASS|SUB-CLA|KPS|BHS|IMG|

    QDEEP_OBJECT_DETECT_CONFIG_MODEL_AOI_GENERAL_DEFECT_DETECTION = 0
    QDEEP_OBJECT_DETECT_CONFIG_MODEL_AOI_PCB_DEFECT_DETECTION = 1
    QDEEP_OBJECT_DETECT_CONFIG_MODEL_AOI_GAUGE_READER_DETECTION = 2
    QDEEP_OBJECT_DETECT_CONFIG_MODEL_BOAT = 3
    QDEEP_OBJECT_DETECT_CONFIG_MODEL_CODESCAN_RECOGNITION_BARCODE = 4
    QDEEP_OBJECT_DETECT_CONFIG_MODEL_CODESCAN_RECOGNITION_QRCODE = 5
    QDEEP_OBJECT_DETECT_CONFIG_MODEL_COMPARISON_AUDIO = 6
    QDEEP_OBJECT_DETECT_CONFIG_MODEL_COMPARISON_VIDEO = 7
    QDEEP_OBJECT_DETECT_CONFIG_MODEL_DEPTH_MAP_3D_EX = 8
    QDEEP_OBJECT_DETECT_CONFIG_MODEL_DRIVING_DISTRACTION = 9
    QDEEP_OBJECT_DETECT_CONFIG_MODEL_EDUCATION = 10
```

QDEEP\_OBJECT\_DETECT\_CONFIG\_MODEL\_FACE\_HEAD\_BODY = 11

QDEEP\_OBJECT\_DETECT\_CONFIG\_MODEL\_FACE\_LANDMARK\_5\_KEYPOINTS = 12

QDEEP\_OBJECT\_DETECT\_CONFIG\_MODEL\_FACE\_LANDMARK\_68\_KEYPOINTS = 13

QDEEP\_OBJECT\_DETECT\_CONFIG\_MODEL\_FACE\_LANDMARK\_68\_KEYPOINTS\_3D\_EX = 14

QDEEP\_OBJECT\_DETECT\_CONFIG\_MODEL\_FACE\_LANDMARK\_68\_KEYPOINTS\_FACE\_BEAUTY\_EX = 15

QDEEP\_OBJECT\_DETECT\_CONFIG\_MODEL\_FACE\_LANDMARK\_68\_KEYPOINTS\_MODAL\_ANALYTICS\_EX = 16

QDEEP\_OBJECT\_DETECT\_CONFIG\_MODEL\_FLAME = 17

QDEEP\_OBJECT\_DETECT\_CONFIG\_MODEL\_HAND\_LANDMARK\_21\_KEYPOINTS = 18

QDEEP\_OBJECT\_DETECT\_CONFIG\_MODEL\_HUMAN\_BACKGROUND\_BLURRING = 19

QDEEP\_OBJECT\_DETECT\_CONFIG\_MODEL\_HUMAN\_BACKGROUND\_REMOVAL = 20

QDEEP\_OBJECT\_DETECT\_CONFIG\_MODEL\_HUMAN\_EPTZ\_AUTO\_FRAMING = 21

QDEEP\_OBJECT\_DETECT\_CONFIG\_MODEL\_HUMAN\_EPTZ\_FACE\_LAYOUT = 22

QDEEP\_OBJECT\_DETECT\_CONFIG\_MODEL\_HUMAN\_EPTZ\_SPEAKER\_TRACKING = 23

QDEEP\_OBJECT\_DETECT\_CONFIG\_MODEL\_HUMAN\_HANDWRITE\_EXTRACTION = 24

QDEEP\_OBJECT\_DETECT\_CONFIG\_MODEL\_HUMAN\_SAFETY\_INSPECTION = 25

QDEEP\_OBJECT\_DETECT\_CONFIG\_MODEL\_HUMAN\_SKELETON\_17\_KEYPOINTS = 26

QDEEP\_OBJECT\_DETECT\_CONFIG\_MODEL\_HUMAN\_SKELETON\_136\_KEYPOINTS = 27

QDEEP\_OBJECT\_DETECT\_CONFIG\_MODEL\_LICENSE\_PLATE\_RECOGNITION\_PARKING = 28

QDEEP\_OBJECT\_DETECT\_CONFIG\_MODEL\_LICENSE\_PLATE\_RECOGNITION\_LAW\_ENFORCEMENT = 29

QDEEP\_OBJECT\_DETECT\_CONFIG\_MODEL\_MISSING\_OBJECT = 30

QDEEP\_OBJECT\_DETECT\_CONFIG\_MODEL\_OPTICAL\_CHARACTER\_RECOGNITION = 31

QDEEP\_OBJECT\_DETECT\_CONFIG\_MODEL\_PHOTO\_CAPTION = 32

QDEEP\_OBJECT\_DETECT\_CONFIG\_MODEL\_PHOTO\_RETOUCHING = 33

QDEEP\_OBJECT\_DETECT\_CONFIG\_MODEL\_PHOTO\_SUPER\_RESOLUTION = 34

QDEEP\_OBJECT\_DETECT\_CONFIG\_MODEL\_RETAIL\_PRODUCT\_RECOGNITION = 35

QDEEP\_OBJECT\_DETECT\_CONFIG\_MODEL\_SECURITY\_TAIWAN = 36

QDEEP\_OBJECT\_DETECT\_CONFIG\_MODEL\_SEGMENTATION = 37

```
QDEEP_OBJECT_DETECT_CONFIG_MODEL_SPEECH_TRANSCRIBE_EN = 38
QDEEP_OBJECT_DETECT_CONFIG_MODEL_SPEECH_TRANSCRIBE_JP = 39
QDEEP_OBJECT_DETECT_CONFIG_MODEL_SPEECH_TRANSCRIBE_ZH = 40
QDEEP_OBJECT_DETECT_CONFIG_MODEL_TRAFFIC = 41
QDEEP_OBJECT_DETECT_CONFIG_MODEL_TRAFFIC_TAIWAN_04 = 42
QDEEP_OBJECT_DETECT_CONFIG_MODEL_TRAFFIC_TAIWAN_08 = 43
QDEEP_OBJECT_DETECT_CONFIG_MODEL_UNATTENDED_OBJECT = 44
QDEEP_OBJECT_DETECT_CONFIG_MODEL_VOICE_CONTROL_EN = 45
QDEEP_OBJECT_DETECT_CONFIG_MODEL_XRAY_INSPECTION_SYSTEM = 46
QDEEP_OBJECT_DETECT_CONFIG_MODEL_CUSTOMIZED = 47
QDEEP_OBJECT_DETECT_CONFIG_MODEL_CUSTOMIZED_LITE = 48
QDEEP_OBJECT_DETECT_CONFIG_MODEL_CUSTOMIZED_MEDICAL_GRADE = 49
QDEEP_OBJECT_DETECT_CONFIG_MODEL_CUSTOMIZED_MULTI_LABELS = 50
QDEEP_OBJECT_DETECT_CONFIG_MODEL_GENAI_EVERYTHING_DETECTION = 51
QDEEP_OBJECT_DETECT_CONFIG_MODEL_GENAI_EVERYTHING_SEGMENTATION = 52
QDEEP_OBJECT_DETECT_CONFIG_MODEL_NVIDIA_CLARA_AGX = 53
QDEEP_OBJECT_DETECT_CONFIG_MODEL_YOYO_V5 = 54
QDEEP_OBJECT_DETECT_CONFIG_MODEL_YOYO_V7 = 55
QDEEP_OBJECT_DETECT_CONFIG_MODEL_RESERVED = 56
```

### 3.4 Confirm the AI Model Build

To build an Ai Model deployment environment, you need to have a corresponding model file and decompress it in the specified path. At the same time, modify the Python file, reference the database path and set the USB Webcam ID. The model can be used normally.



1. Add it to the Python development program according to the general parameters listed above.
2. Unzip the corresponding model (7z file) to the corresponding folder path.
3. Unzip the corresponding resource file and add environment parameters (the following instructions apply).

Enter the following commands, check whether the file exists, and unzip the corresponding resource file and add environment parameters.

```
# unzip qdeep/lib.zip
```

```
# export LD_LIBRARY_PATH=./lib
```

## 4 PRECAUTIONS AND EXCEPTION ELIMINATION

Prompts the exceptions or precautions during installation after the model is installed, and provides the methods and instructions for elimination.

### 4.1 Ubuntu

1. Do not update the Ubuntu system by yourself or use instructions to update or upgrade.
2. Only update (get the APT repository package list update).
3. Check whether the device supports Linux when using USB WebCam.
4. It is recommended to use the TYPE C Hub to augment the use of mouse and keyboard and connect to WebCam.

### 4.2 Ubuntu Suite

1. Check the python version on the system side :

```
Python3 -V
```

2. Confirm/install the numpy kit :

```
sudo apt-get install python3-numpy °
```

3. Install the V4L2-CTL tool to check the status of the video device and confirm the relevant information. :

```
sudo apt install v4l-utils
```

4. Install GStreamer related kits :

```
sudo apt-get install libgstreamer1.0-dev gstreamer1.0-plugins-base  
gstreamer1.0-plugins-good gstreamer1.0-plugins-bad gstreamer1.0-  
plugins-ugly °
```

### 4.3 Exception Error Handling

1. OpenBLAS

- **Error Message :**

```
libopenblas.so.0: cannot open shared object file:
```

```
No such file or directorySegmentation fault (core dumped)
```

- **libopenblas.so.0: cannot open shared object file: No such file or directory :**
- **The system could not find libopenblas.so.0, a shared library file.**

This is usually because the OpenBLAS library is not installed, is installed in the wrong place, or is missing from the system's library search path.

- **Segmentation fault (core dumped) :**

The program crashes when trying to access an area of memory that is not allowed. This is usually due to missing or incompatible dependent libraries that prevent the program from working properly.

- **Exclusion Method :**

Depending on the system, the method of installing OpenBLAS may vary. Let's take Ubuntu or Debian as an example :

For Ubuntu/Debian systems :

Open Terminal and execute the following command to install OpenBLAS :

```
sudo apt update
```

```
sudo apt install libopenblas-base libopenblas-dev
```

- libopenblas-base : Provides the OpenBLAS library at runtime.
- libopenblas-dev : Provides header files and library files for development.

## 2. v4l2-ctl query USB Webcam device

- **Permission issues :**

- Make sure you have permission to access the /dev/video\* device. You can use sudo to elevate privileges :

```
sudo v4l2-ctl -d /dev/video0 --all
```

- **The device does not exist or is occupied :**

- Make sure that the specified video device exists and isn't occupied by other apps.
- Use ls /dev/video\* to confirm the presence of the device.

- Use `ls /dev/video0` to check if there are any other programs occupying the device.

- **Driver issues :**

- Make sure that the drivers for the video device are properly installed and set up successfully.
- Use `dmesg | grep video` or `dmesg | grep uvcvideo` to see the driver loading status and related error messages.

### 3. Unzip the Exception 7z kit

- **Install the p7zip kit using apt :**

The related suite names for 7-Zip are `p7zip` and `p7zip-full`. You can install these kits with the following command :

```
sudo apt update
```

```
sudo apt install p7zip-full p7zip-rar
```

This command installs :

- **p7zip-full :**

Contains the full version of 7-Zip that supports compression and decompression in 7z format.

- **p7zip-rar :**

Supports file decompression in RAR format.

### 4. No module named 'cv2' error

This error indicates that the OpenCV module is not installed in the Python environment (`cv2` is the Python equivalent of OpenCV), and the following are the steps to resolve this issue.

- **Install the OpenCV module :**

Install the OpenCV module using pip :

```
pip install opencv-python
```

If you need full functionality, including video processing, install the following packages :

```
pip install opencv-python-headless opencv-contrib-python
```

- **Verify that the installation was successful :**

Once installed, go into the Python interactive shell and try to import cv2 :

```
import cv2
```

```
print(cv2.__version__)
```

If there are no errors and you can see the version number, the installation was successful.

5. **OSError: libnppicc.so.11: cannot open shared object file: No such file or directory error**

The NVIDIA CUDA toolkit is missing, please enter the following command to install.

```
sudo apt install nvidia-jetpack
```

## 5 GITHUB FILE DOWNLOAD AND DECOMPRESSION

Download the specified project file from GitHub, unzip them one by one according to the instructions above, and build the project directory and SDK to complete the AI Model project establishment.

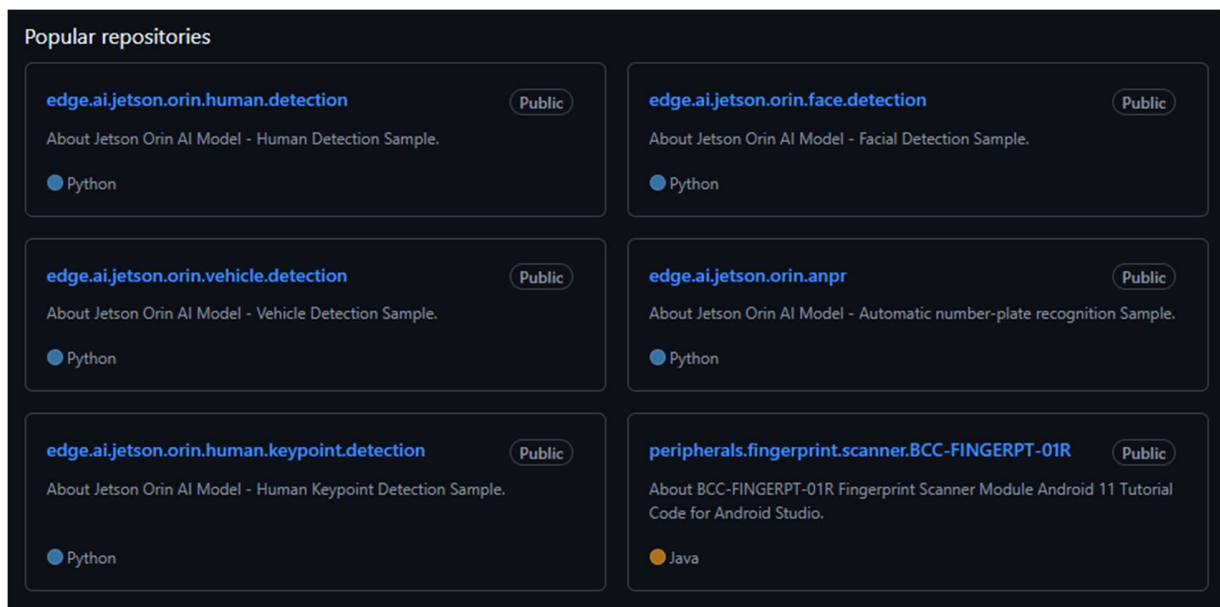
For Ai Model project files, since you use Git LFS to upload files, it is recommended to use Git Cmd to download them through commands to avoid the model files being unusable after downloading.

### 5.1 GitHub file download (example)

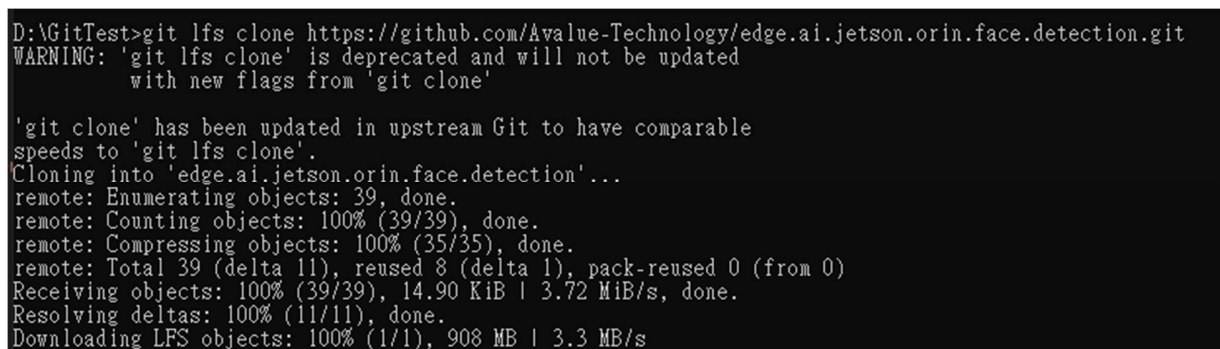
1. Download through the GitHub link of each AI model and use Git Cmd to input the following command for downloading.

The command is as follows:

Git lfs clone <https://github.com/Avalue-Technology/edge.ai.jetson.orin.face.detection.git>




(GitHub Directory)



(GitCmd Example image)

Notice: After downloading the AI project files, please check the directory for the presence of the xxx.7z file and confirm whether the file size is correct. If the file size is only 1KB, it is likely tha

t the download was incorrect (as shown in Figure 1); if the file is downloaded correctly, the file size should be greater than 70MB (as shown in Figure 2).

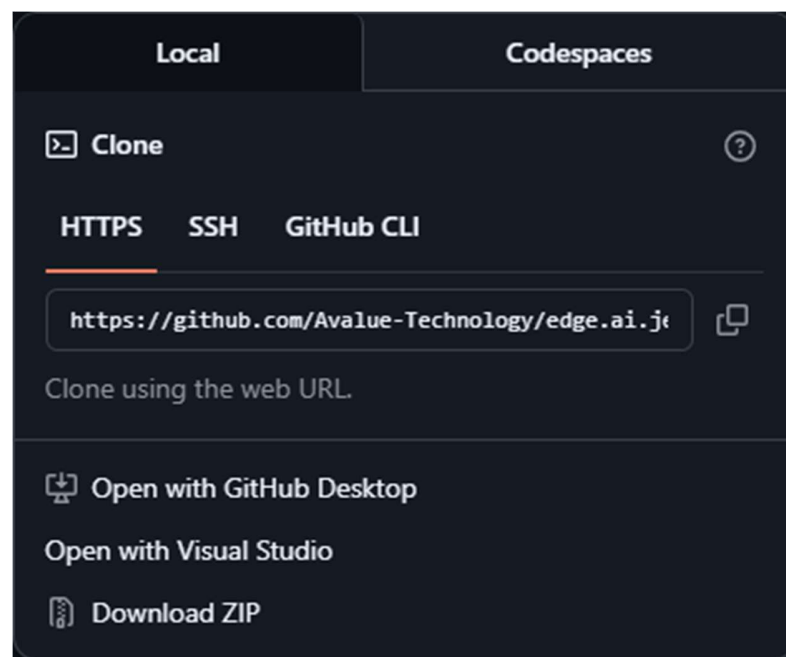
名稱	修改日期	類型	大小
 .gitattributes	2025/4/30 下午 02:21	文字文件	1 KB
 RELEASES.QDEEP.MODEL.FACE.LANDMARK.5KPS.7z	2025/4/30 下午 02:21	7Z 檔案	1 KB

(Figure 1: Abnormal 7z file)

名稱	修改日期	類型	大小
 .gitattributes	2025/4/30 下午 04:42	文字文件	1 KB
 RELEASES.QDEEP.MODEL.FACE.LANDMARK.5KPS.7z	2025/4/30 下午 04:47	7Z 檔案	887,050 KB

(Figure 2: Correct 7z file)

### 2. Download AI Model ZIP File (click on Download ZIP)

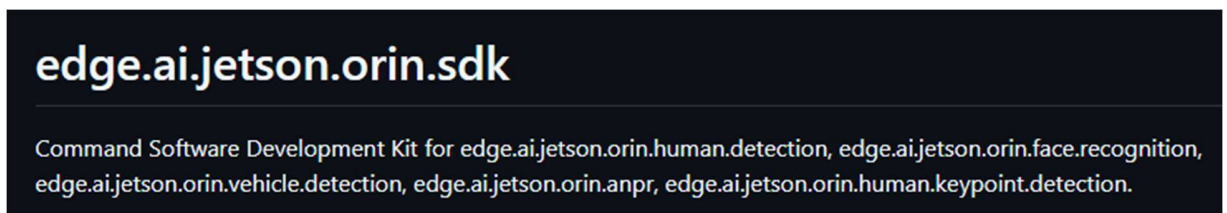


(Figure 3: GitHub download link)

### 3. Download completed, extract the corresponding files

名稱	修改日期	類型	大小
model	2025/4/30 下午 02:21	檔案資料夾	
Python File	2025/4/30 下午 02:21	檔案資料夾	
LICENSE	2025/4/30 下午 02:21	檔案	2 KB
README.md	2025/4/30 下午 02:21	Markdown 來源...	1 KB

- Download the corresponding SDK file



(Figure 4: SDK GitHub Directory)

- The SDK and AI model files are transferred from the storage device to the device machine and the corresponding directories are created

model	2025/4/30 下午 04:42	檔案資料夾	
Python File	2025/4/30 下午 04:42	檔案資料夾	
SDK	2025/5/6 下午 04:08	檔案資料夾	
LICENSE	2025/4/30 下午 04:42	檔案	2 KB
README.md	2025/4/30 下午 04:42	Markdown 來源...	1 KB

Move the relevant SDK and Python file to the directory.

RELEASES.QDEEP.MODEL.FACE.LAND...	2025/4/30 下午 04:47	7Z 檔案	887,050 KB
README.md	2025/4/30 下午 04:42	Markdown 來源...	1 KB
camera_face.py	2025/4/30 下午 04:42	Python 來源檔案	15 KB
qcap.zip	2025/4/30 下午 02:44	壓縮的 (zipped) ...	30,068 KB
qdeep.zip	2025/4/30 下午 02:44	壓縮的 (zipped) ...	93,377 KB
LICENSE	2025/4/30 下午 04:42	檔案	2 KB

- Enter the command below to decompress and create a directory

- ✓ unzip qdeep/lib.zip
- ✓ 7z e RELEASES.QDEEP.MODEL.FACE.LANDMARK.5KPS.7z -omodel/ FACE\_LANDMARK\_5KPS



✓ `export LD_LIBRARY_PATH=./lib`

### 7. Execute Python File

✓ `python3 camera_person.py`