Baidu Brain EdgeBoard AI Box &Accelerator Card(FZ5) User Manual
V1.0

Version Change History

Version	Description	Date
V1.0	Initial Version	2020/08/05

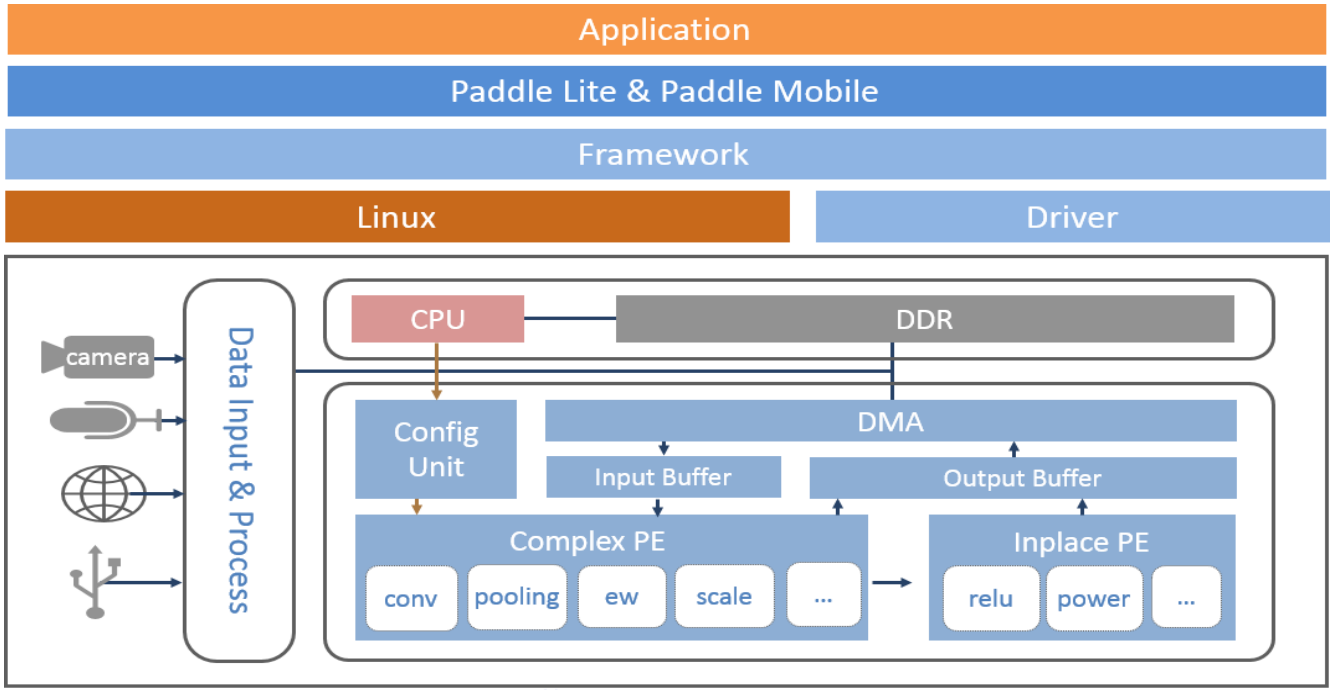
Table of content

Introduction

MYIR's FZ5 is a deep learning computing card based on Xilinx Zynq Ultrascale+ CZU5EV MPSoC, which is closely cooperated with Baidu. It integrates a 4-core ARM A53 processor + GPU + FPGA architecture, with multi-core processing capabilities, FPGA Programmability and video streaming hardware decoding capabilities. It has built-in deep learning soft core based on Linux operating system + Baidu deep learning platform Paddle, deeply compatible with Baidu brain model resource and tool platform (EasyDL/AIStudio), can effectively and quickly implement a series of processes such as model training-deployment-inference, which greatly reduces the threshold of development verification, product integration, scientific research and teaching, project deployment.

1. Introduction to soft cores

1.1 Software Introduction



The FZ5 computing card comes with a Linux OS, users can develop applications based on the Linux OS. (Main calling process: 1. The application obtains video input -> 2. Calls the prediction library to load the model -> 3. The scheduling model and the underlying driver acceleration module are used for the calculation -> 4. Get the running result).

1.2 Performance data on FZ5 for common models (unquantified cropping)

Network	Input Size	Single frame time consumption
resnet50	224 x 224	42ms
mobilenet-v1	224 x 224	10ms
inception-v2	299 x 299	41ms
inception-v3	299 x 299	70ms
resnext	224×224	69ms
mobilenet-ssd	300 x 300	24ms
mobilenet-ssd-640	640 x 640	79ms
vgg-ssd	300×300	246ms
yolov3	608×608	582ms

Note: FZ5 soft core is still under upgrading, and its performance will be improved simultaneously. Different versions of the same network structure have different computing power requirements. If there are specific project applications, you can contact the official team to apply for customized optimization

2. Start and connect

2.1 Start-up preparation

1. The default start-up mode is start-up from TF card. In order to use the functions of the device normally, please plug the TF card into the TF card slot.
2. The board supports serial port debugging and network port debugging. Serial port debugging corresponds to the equipment's console, it may have redundant printing information during general debugging, so network port debugging is recommended. The board's static IP is pre-set: 192.168.1.254, users can connect the board to the computer directly or to routing devices through network cable (Network cable needs to be prepared by user), using the SSH protocol to log in to the system. Please refer to the following article for specific usage.
3. Username and password (root and root) are required to be entered after the board starts-up to login in the OS. The OS has its own deep learning pre-installed environment as well as model inference sample. Please refer to the following article for detailed information.

2.2 Connection Method One: SSH Connection

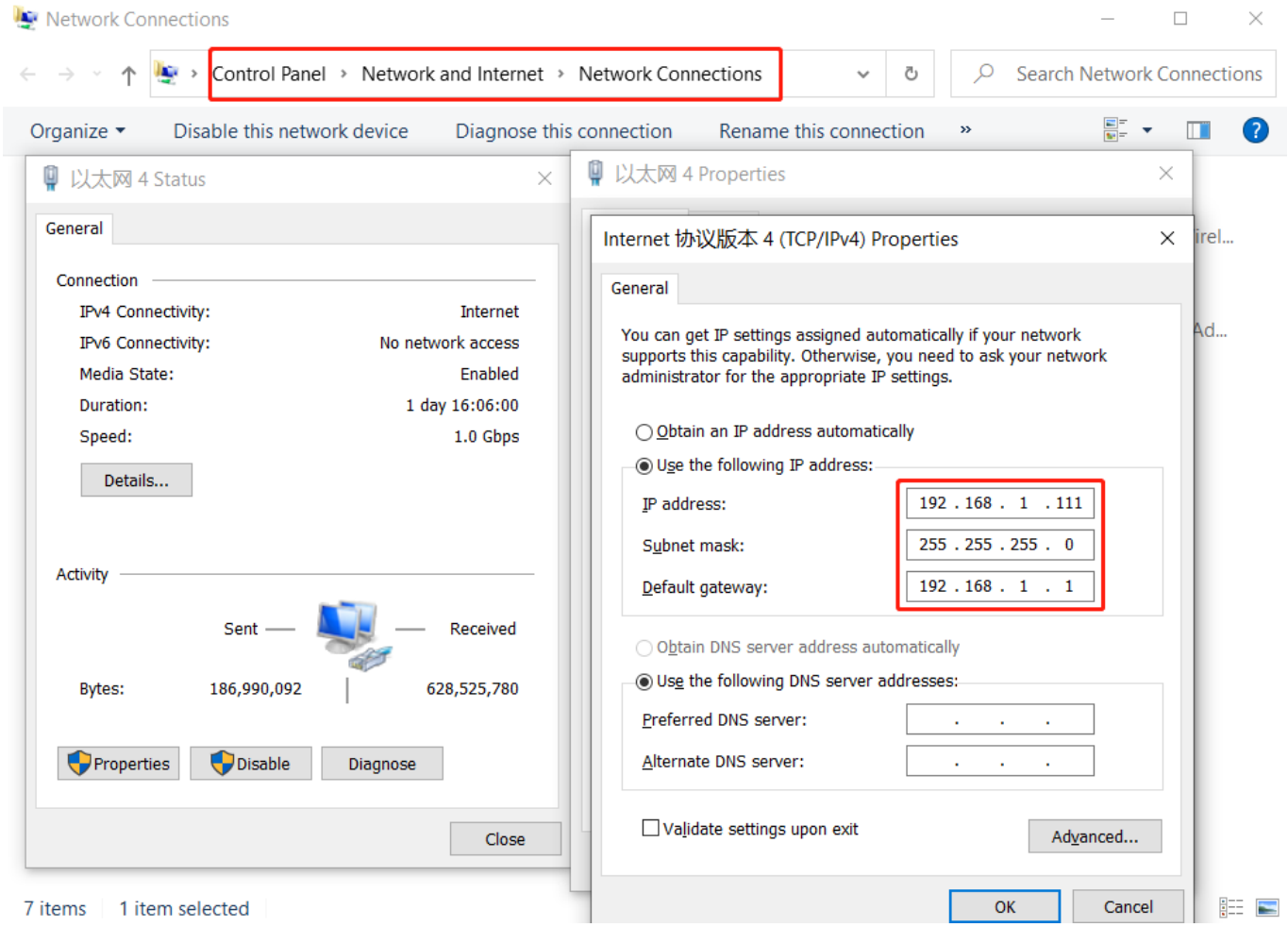
- FZ5's default parameter: static IP=192.168.1.254, netmask=255.255.255.0, gateway=192.168.1.1
- Hardware connection method: Connect FZ5 to the host computer or router, set the computer or router IP to the same segment as FZ5, then users can login in via SSH.

Detailed Steps：

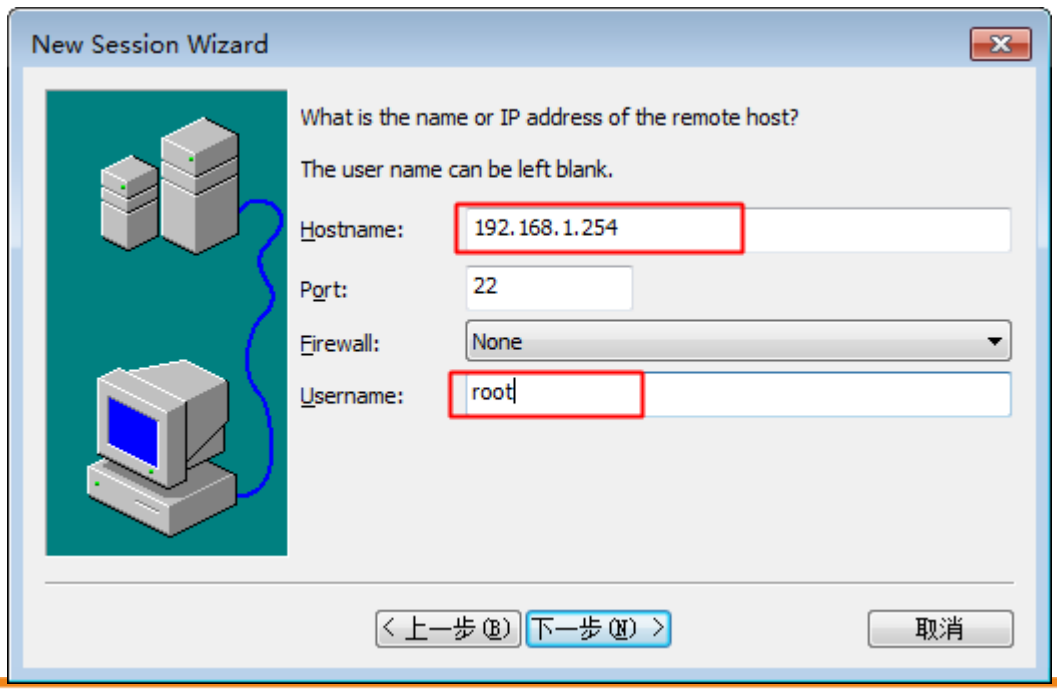
2.2.1 How to connect to the internet via SSH in Windows

1. Install debugging tools, SecureCRT (Users may Google search and install) is recommended.

2. Set IP of the PC or router to the same network segment with the board. When the PC and the board is connected directly, IP of the computer needs to be set manually: open the Control Panel->Network and Internet->Network Connections->Local Connection->Properties-> The Internet protocol version 4, set the IP address manually: IP address=192.168.1.111, Subnet mask=255.255.255.0, Default gateway=192.168.1.1 as shown in below picture:



3. Create new window in SecureCRT. Click connect->New session->Protocol(select SSH2), click next, Hostname is default IP of FZ5: 192.168.1.254, Port is 22, Username is root, password is root,



then click next to login in.

2.2.2 How to connect to the internet via SSH in MAC

1. Set IP of the PC or router to the same network segment with the board.

Detailed steps：System preferences-->Network-->Advanced-->TCP/IP。

Ipv4 configuration example: Manual, Ipv4 address:192.168.1.111, Subnet mask: 255.255.255.0, Router: 192.168.1.1.

2. Open Terminal: Launchpad->Other(folders)->Terminal

3. Enter ssh root@ip (IP of FZ5) in the Terminal: the default parameter is ssh root@192.168.1.254, then enter username and password (root and root) to login in.

2.3 Connection Method Two: Serial Port Connection

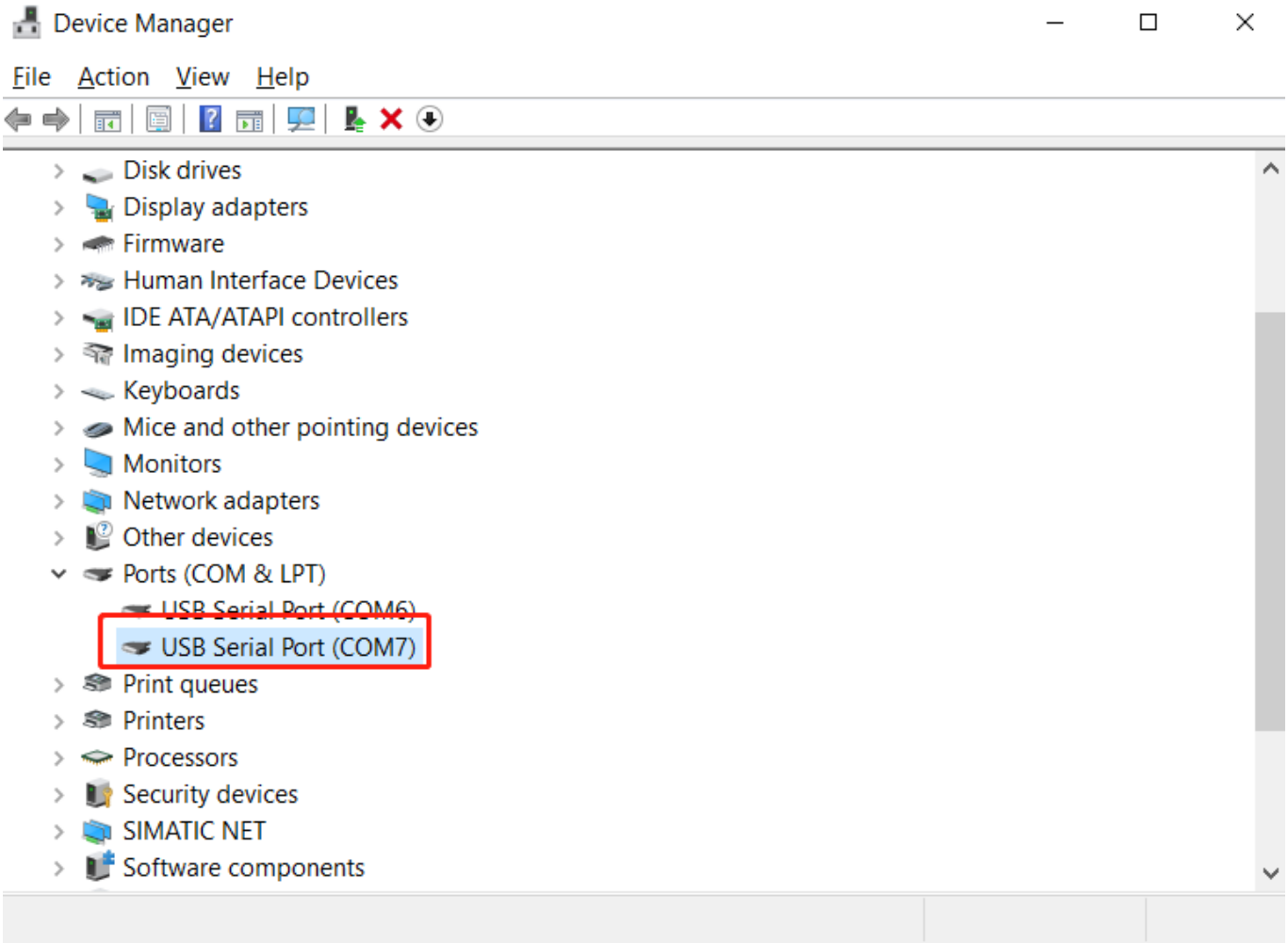
If SSH connection fails or the IP needs to be viewed (after IP of the board is dynamically obtained), the serial port needs to be used to enter the console of the board.

Connect USB UART of FZ5 to PC.

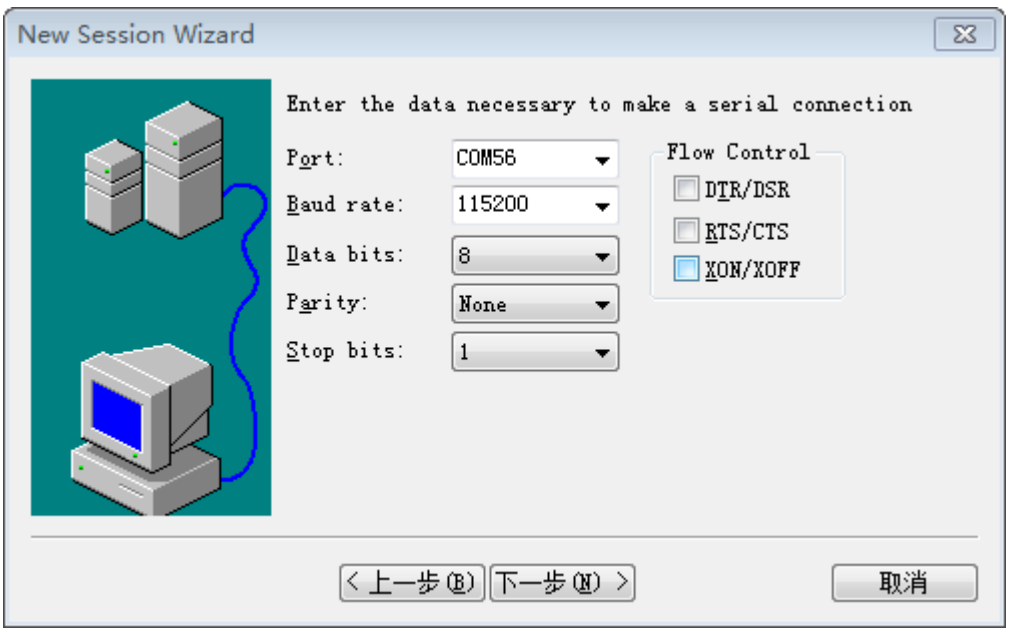
2.3.1 Method of using SecureCRT to connect serial port in Windows

1. Install software SecureCRT and serial driver cp210x_Windows_Drivers (The driver needs to be installed for the first time, and the installation package can be searched from Google)

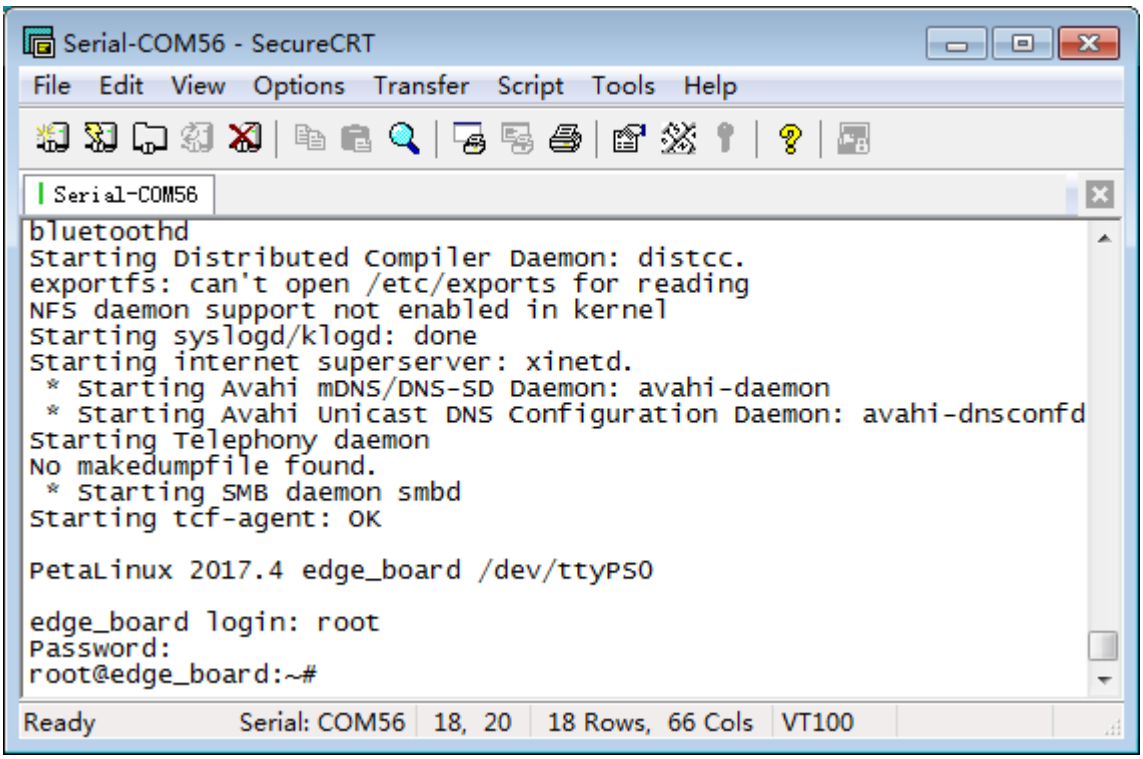
2. Make sure that the PC is connected to the USB UART interface of FZ5, and then click "my computer" -> Properties -> device manager to check the port number mapped in the device manager. As shown in below picture, the port number is com7.



3. Open SecureCRT, create a new window Connect -> New Session -> Protocol, select serial, baud rate 115200, not selected flow control, as shown in the picture below



4. Click [finish] and [connect] button, SecureCRT will connect to the serial port on the computing card. After power on, users can see the start-up information. After the startup is completed, enter the user name and password to enter the board's OS. As shown in below picture.



2.3.2 Method of using Minicom to connect serial prot in Mac OS

1. Install the serial port driver SiLabsUSBDriverDisk.dmg (it needs to be installed for the first time, users may Google to search it) into PC
2. Install the Minicom tool
3. **Open Terminal:** Launchpad->Others->Terminal, input command “minicom -s” to configure.
4. The configuration content is as below. After configuration, connect FZ5 and input minicom in terminal

- select Serial port setup :
  A - Serial Device : /dev/cu.SLAB_USBtoUART
  B - Lockfile Location :
  /usr/local/Cellar/minicom/2.7/var C - Callin
  Program :
  D - Callout Program :
  E - Bps/Par/Bits : 115200
  8N1 F - Hardware Flow
  Control : No G - Software
  Flow Control : No
- Save setup and dfl

3. Debug The device

3.1 Change the network configuration

The default IP address of the board is 192.168.1.254. If multiple boards are connected to the same LAN at the same time, their IP addresses need to be configured different or changed to dynamically obtaining IP. The path of network configuration file is /etc/network/interfaces

```
//Open and edit the interfaces file
vim /etc/network/interfaces
```

Static IP configuration

```
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet static
address 192.168.1.254
netmask 255.255.255.0
gateway 192.168.1.1
broadcast 192.168.1.255
```

Dynamic IP configuration

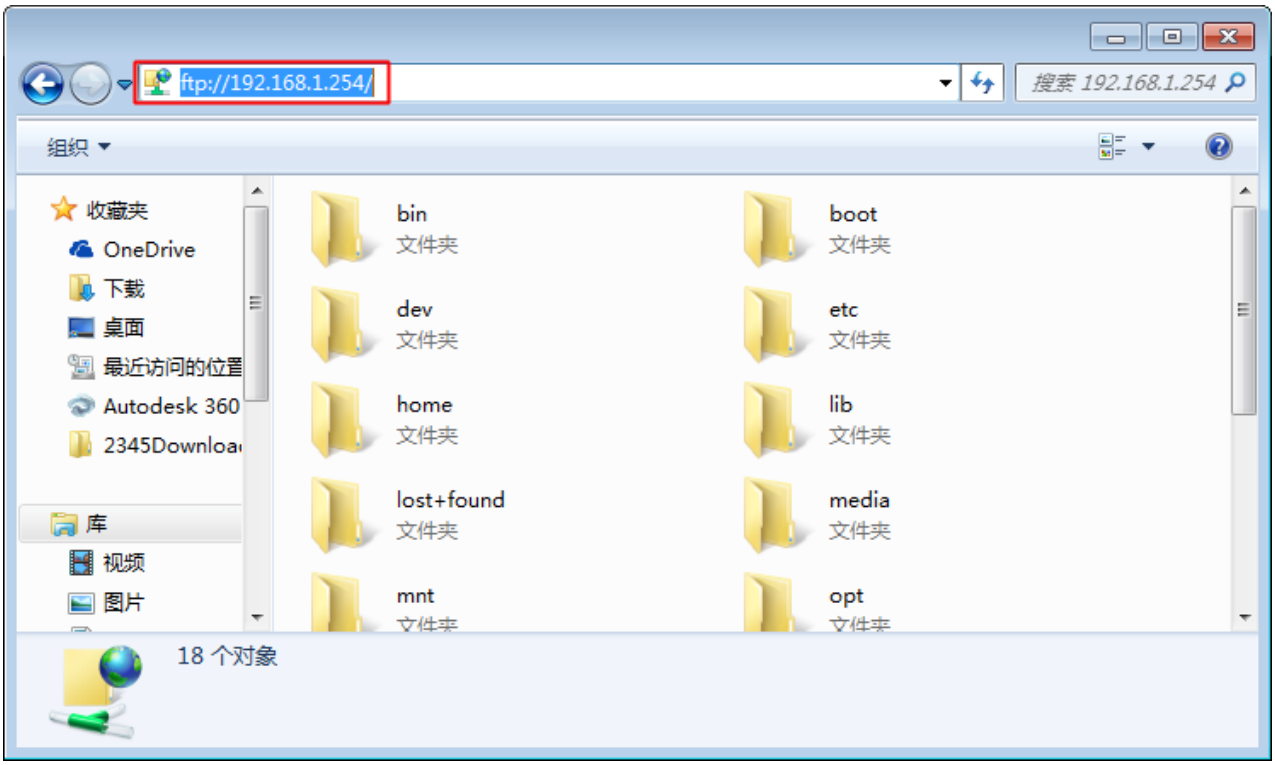
```
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet dhcp
```

3.2 Copy files

FZ5 supports SSH, samba, FTP and other network protocols, can easily carry out data communication and file copy through the network. This function will be widely used in software upgrading and user customization.

3.2.1 Realize File copy through FTP (suitable for Windows system)

1. [Press Windows + R shortcut key combination, input ipconfig], check the board's IP by using ipconfig command, ensure that the board's IP and the Windows PC's IP are in the same network segment. Input ftp://192.168.1.254 in the folder input box. Enter the user name root and password root according to the prompt, then users can enter the equipment system.



2. Open home -> root -> workspace directory. Workspace is the directory where the application is located under the user root. Then copy the file to workspace, or copy files from workspace to PC.

3.2.2 Realize file copy through Samba protocol (For Mac OS)

1. Configure and ensure that the board's IP and MAC's IP are in the same network segment (see above 2.2.2: How to connect to the internet via SSH in MAC)
2. After the configuration is completed, click Finder -> To ->Connect to the server, enter smb://ip, for example smb://192.168.1.254, user name root, password root.
3. The file directory of the board appears in "Finder". Open home -> root -> workspace directory. Workspace is the directory where the application program is located under the root user. Users can copy the files between the PC and the board through the copy and paste command.

3.3 Introduction to system catalog

Content	Catalog	Remarks
paddle-mobile	/home/root/workspace/paddle-mobile	paddle-mobile prediction Library
Driver	/home/root/workspace/driver	Device drivers
sample	/home/root/workspace/sample	Sample codes
tools	/home/root/workspace/tools	Debug tools

3.4 Run Samples

When user's PC is connected to FZ5, user can run the deep learning examples which MYIR provides. It locates in /home/root/workspace/sample.

Sample	Remarks
classification	Example of classification model
detection	Example of target detection model

3.4.1 Example of Classification model

Read a local image, call the model for reasoning, and output the results.
Considering the simplicity and generality, this example shows reading the model and image information from the JSON file, then loads and executes it. When executing, users need to specify the corresponding configuration file:

```
./image_classify ../configs/resnet50/drink.json
```

The structure of project:

```
├── CMakeLists.txt //cmake configuration file
├── include
|   └── io // paddle_mobile //Content of head files
|       ├── paddle_inference_api.h
|       ├── type_define.h
|       └── types.h
├── configs //Content of configuration files
|   ├── Inceptionv2
|   |   └── zebra.json //Configuration of zebra pictures
```

```
|   ├── Inceptionv3
|   |   └── zebra.json
|   ├── mobilenetv1
|   |   └── zebra.json
|   └── resnet50
|       └── drink.json
├── lib
|   └── libpaddle-mobile.so
├── models // Content of models
|   ├── Inceptionv2
|   ├── Inceptionv3
|   ├── mobilenetv1
|   └── resnet50
|── src
|   ├── json.hpp // json lib
|   ├── video_classify.cpp // Example of video reasoning
|   └── image_classify.cpp // Example of picture reasoning
└── README.md
```

Here is an example of a configuration file.

```
{
    "model":"../models/resnet50",
    "combined_model":true, //
    "input_width":224,
    "input_height":224,
    "image":"../models/resnet50/drink.jpeg",
    "mean":[104,117,124],
    "scale":1,
    "format":"BGR"
}
```

key	value
model	Directory of model
combined_model	Whether it is a fusion model or not, only two files are fusion models
input_width	Enter the image size for neural network. Following input images will be scaled to this size
input_height	Enter the image size for neural network.
image	Image input for classification
mean	Average value
scale	Before entering the neural network, the budget processing is (x - mean) * scale
format	The format required by the neural network, Opencv is BGR by default

Other classification networks can also be implemented by adding/modifying configuration files without modifying the code.

Example of detailed steps:

1. Load driver, after the system starts, it can be loaded automatically or manually (It has been set loaded automatically by default when leaving factory)

```
insmod /home/root/workspace/driver/fpgadrv.ko
```

2. Compile the example. FZ5 has the ability to compile. Go to the directory build of the sample/classification to compile

```
// Open the sample directory
cd /home/root/workspace/sample/classification
// If there have no build directory, create onemkdir build cd build
rm -rf *
// Call cmake to create Makefile
cmake ..
// Compile
make
```

After compiling, the following files will be generated in the directory build:

image_classify: To read local image reasoning example。

video_classify: To read the camera data for reasoning, which can only be used by connecting the USB camera or the HDMI video source. To display the results, connect display device via the DP interface (DP can also be transferred to HDMI or VGA).

For some industrial application scenarios that need to use HDMI as the video source input interface, we provide a HDMI IN interface. The system, software and hardware have been adapted to each other and can be used directly.

3.Run sample

```
Using picture reasoning
./image_classify ../configs/resnet50/drink.json

Using video reasoning, it is necessary to ensure that the USB camera and the monitor are connected at the same time
startx  // Open desktop environment
./video_classify ../configs/resnet50/drink.json
```

3.4.2 Example of object detection

Different from classification, object detection can not only detect the type of the object, but also detect the location coordinates of the object. There are two examples for object detection. One is to detect objects on the image and draw coordinate information. The other one is the camera collects video and detects the coordinates information, then the information would be drawn on the screen.

Project directory structure:

```
├── CMakeLists.txt。
├── include
│   └── io // paddle_mobile
│       ├── paddle_inference_api.h
│       ├── type_define.h
│       └── types.h
```

```
├── configs // profile directory
│   ├── mobilenet-ssd
│   │   └── screw.json // Picture configuration of screws and nuts
│   ├── mobilenet-ssd-640
│   │   └── screw.json
│   └── vgg-ssd
│       └── screw.json
├── lib
│   └── libpaddle-mobile.so
├── models
│   ├── mobilenet-ssd
│   ├── mobilenet-ssd-640
│   └── vgg-ssd
├── src
│   ├── json.hpp
│   ├── video_detection.cpp
│   └── image_detection.cpp
└── README.md
```

Below is an example of configuration file.

```
{
    "model":"../models/vgg-ssd",
    "combined_model":true, //
    "input_width":224,
    "input_height":224,
    "image":"../models/vgg-ssd/screw.jpg",
    "mean":[104,117,124],
    "scale":1,
    "format":"BGR"
}
```

key	value
model	Directory of model
combined_model	Whether it is a fusion model or not, only two files are fusion models
input_width	Enter the image size for neural network. Following input images will be scaled to this size
input_height	Enter the image size for neural network.
image	Image input for classification
mean	Average value
scale	Before entering the neural network, the budget processing is (x - mean) * scale
format	The format required by the neural network, Opencv is BGR by default
threshold	Confidence threshold

Other classification networks can also be implemented by adding/modifying configuration files without modifying the code.

Example of detailed steps:

1. Load drive, after the system starts, it can be loaded once (the factory has automatically loaded by default)

```
insmod /home/root/workspace/driver/fpgadrv.ko
```

2. Compile the example. FZ5 has the ability to compile. Go to the directory build of the sample/classification to compile

```
cd /home/root/workspace/sample/detection

mkdir build
cd build
rm -rf *

cmake ..

make
```

After compiling, the following files will be generated in the directory build:

image_classify: To read local image reasoning example。

video_classify: To read the camera data for reasoning, which can only be used by connecting the USB camera or the HDMI video source. To display the results, connect display device via the DP interface (DP can also be transferred to HDMI or VGA).

3.Run sample

```
// Using picture reasoning
./image_detection ../configs/vgg-ssd/screw.json

// Using video reasoning
startx
./video_detection ../configs/vgg-ssd/screw.json
```

3.4.3 To output reasoning result and display

1. Display via Mini DP interface: Connect FZ5 and DP monitor with male to male mini DP cable.

2. Display via HDMI: Connect FZ5 and HDMI monitor using active mini DP to female HDMI cable.

3. Display via VGA: connect FZ5 and VGA monitor with male mini DP to female VGA cable.

The software package comes with FZ5 includes a simplified version of Linux desktop environment which can be used to display the effect of program running in real time. Please make sure that the monitor and FZ5 have been connected before start-up. After entering the system, the board will enter the terminal command line environment by default. Users may enter and exit the desktop environment using the following commands.

```
startx //Open desktop environment
stopx //Close desktop environment
```

For the demonstration example MYIR provides, opencv supports the visualization of prediction results through window of desktop, please refer to 3.4.1 and 3.4.2 for how to use.

3.4.4 Video input mode

FZ5 supports video data input of USB, HDMI input, GigE and other protocols, can be used as the processing module of video stream in various scenarios.
For some industrial application scenarios that need to use HDMI as the video source input interface, we provide a HDMI IN interface. The system, software and hardware have been adapted to each other and can be used directly.

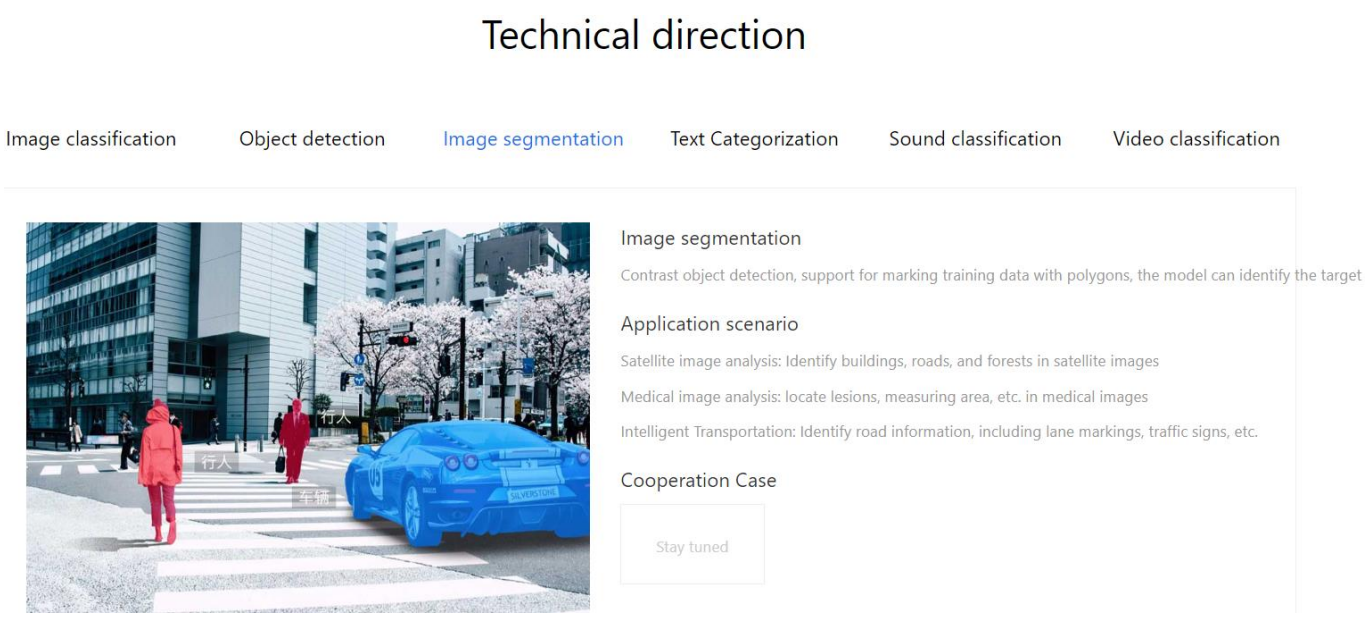
3.5 Running EasyDL platform model prediction example

3.5.1 How to use EasyDL

EasyDL is a one-stop deep learning model training and service platform that provides a visual operation interface. Users can obtain a high-precision model by uploading a small number of pictures. For details, please refer to the EasyDL official website. Detailed steps of using EasyDL for data training is as below.

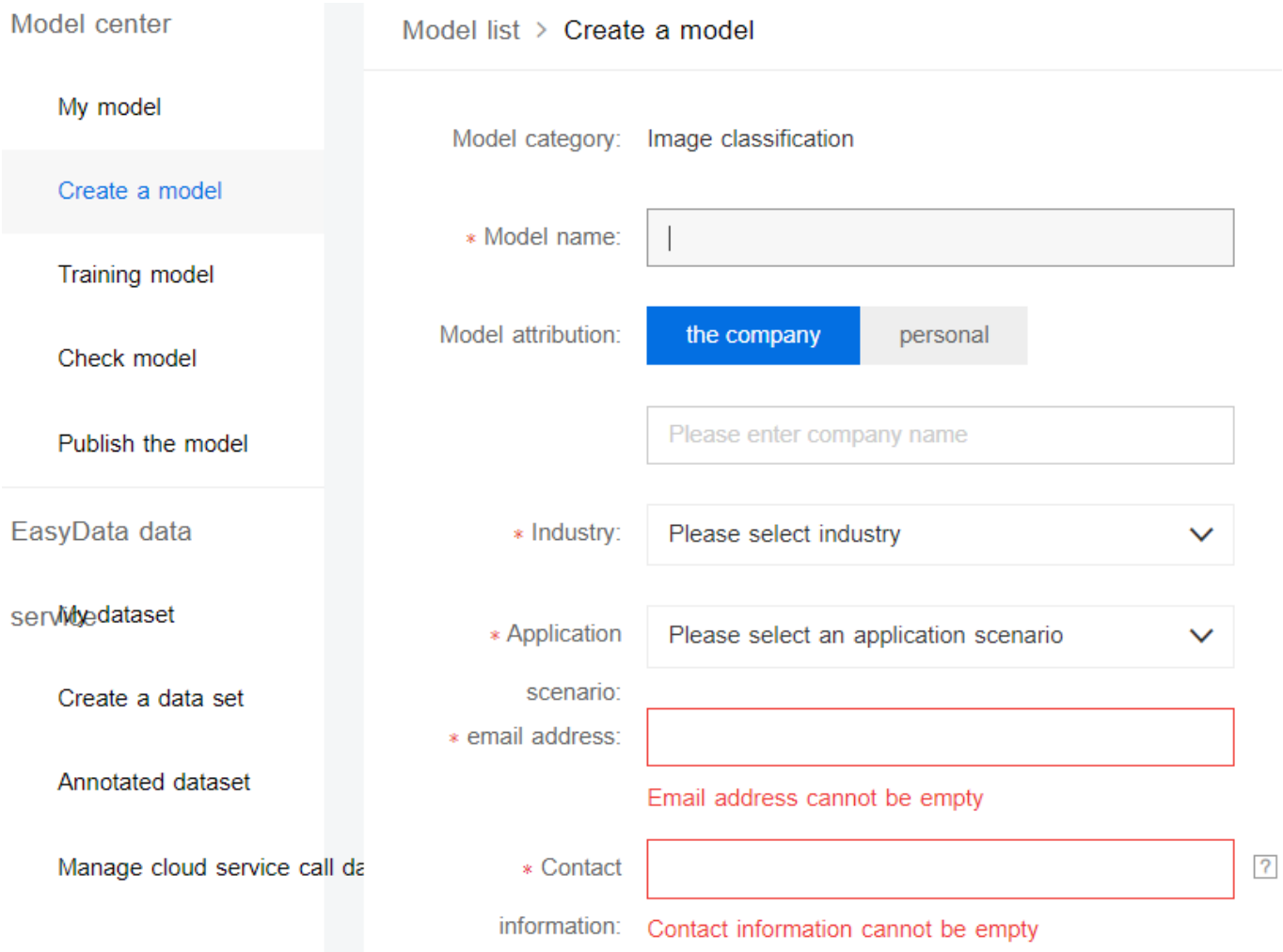
3.5.1.1 Select training category

Users may choose "image classification" or "object detection" according to the general scene.



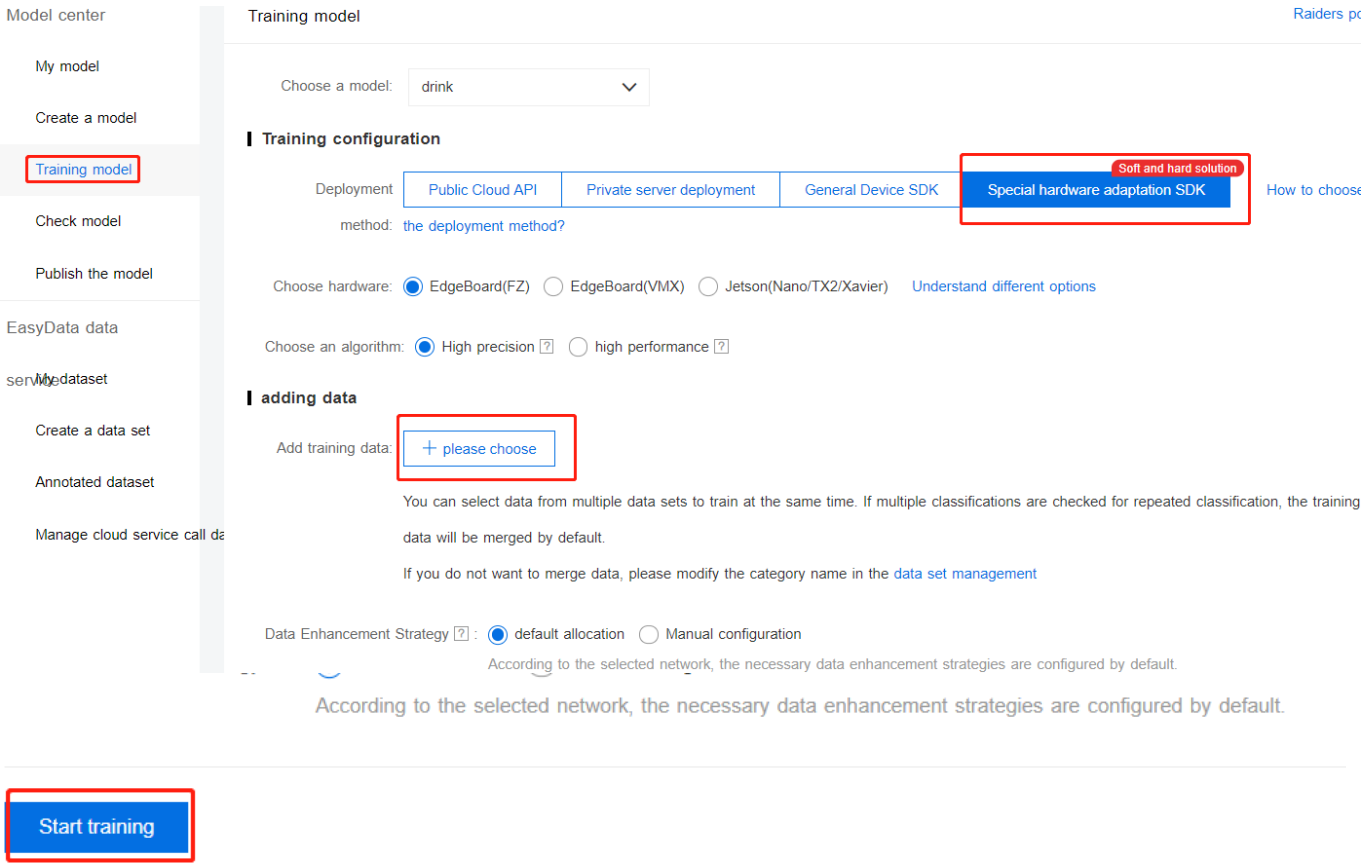
3.5.1.2 Create “Image classification” model

Select “Image classification”->”Train now”->”Create model”, fill in the content of the model according to personal needs, and the * sign is must. Select "next" after completion



3.5.1.3 Training model

Enter “Training model”, select the model category, select “Special hardware adaptation SDK”, must select “EdgeBoard (FZ)”, click “Please choose” to choose training data, after completion, check "add recognition result as other default classification", and select "start training". According to the size of the training data set, the training time will be different. Generally, about 100 pictures can be completed in 10 minutes at most.



3.5.1.4 Generate SDK

1. After the training is completed, apply for the release model in" my model", select the "EdgeBoard + dedicated SDK" in the integrated software and hardware solution, submit the application, and wait for review.

2. After the model review is approved, click the "Service Details" button at "My Model" and select the download SDK in the dialog window that pops up.

3. Get serial number

Click "Manage Serial Number" to skip to Baidu Cloud-->EasyDL customized training platform-->Offline SDK management interface to view the serial number for activating the SDK.

3.5.1.5 Install the SDK in FZ5

1. The downloaded software deployment package contains SDK and demo which are easy to use. FZ5 can be quickly deployed and run after a few simple steps. The deployment package file structure is as below.

```
EasyEdge-m1800-EdgeBoard/
├── cpp
|   └── baidu_easyedge_linux_cpp_aarch64_PADDLEMOBILE_FPGA_v0.3.2_gcc6.2_20190518
|       ├── demo
|       |   ├── CMakeLists.txt
|       |   ├── demo.cpp
|       |   └── easyedge_serving
|       ├──include
|       |   ├── easyedge
|       |       └── easyedge.h
|       └── lib
|           ├── libeasyedge.so -> libeasyedge.so.0.4.0
|           ├── libeasyedge.so.0.4.0
|           ├── libeasyedge_static.a
|           ├── libpaddle-mobile.so
|           └── libverify.alib
├── RES
|   ├── conf.json
|   ├── label_list.txt
|   ├── model
|   ├── params
|   └── preprocess_args.json
|
└── tools
```

2. Use serial number license to activate the SDK

Open the file demo.cpp

【Directory：EasyEdge-m1800-edgeboard/cpp/baidu_easyedge_linux_cpp_aarch64_PADDLEMOBILE_FPGA_v0.3.2_gcc6.2_20190518/demo/demo.cpp】

Write license serial number

```
int main(int argc, char *argv[]) {

    if (argc != 3) {
        std::cerr << "Usage: demo {model_dir} {image_name}";
        exit(-1);
    }

    PaddleFluidConfig config;
    config.model_dir = argv[1];
    global_controller()->set_licence_key("set your license here");
    global_controller()->log_config.enable_debug = false;
    auto predictor = global_controller()->CreateEdgePredictor
<PaddleFluidConfig, EdgeEngineKind::kPaddleMobile>(
            config);
```

3. Put the SDK in the directory /home/root/workspace/ directory (see "file copy" above) for decompression, then start and run according to the following methods.

3.5.2 Run SDK

1. The driver can be loaded once after the system starts-up (Or add the system startup script, please refer to 4.1.3 for details)

```
insmod /home/root/workspace/driver/fpgadrv.ko
```

If the driver is not loaded, the following error may be reported：

```
Failed to to fpga device: -1
```

Set system time (System time must be correct)

```
date --set "2019-5-18 20:48:00"
```

2. Compile

```
cd /home/root/workspace/EasyEdge-m1800-edgeboard/cpp/baidu_easyedge_linux_cpp_aarch64_PADDLEMOBILE_FPGA_v0.3.2_gcc6.2_20190518/demo

mkdir build
cd build
rm -rf *

cmake ..

make
```

3.Run sample

```
// Run the execution file in the build directory
./easyedge_demo {RES Resource folder path} {Test image path }
```

For example: Put the image which you want to predicte in the SDK. If you put the image into the folder RES, you can see the recognition result

```
 ./easyedge_demo /home/root/workspace/EasyEdge-m1800-edgeboard/RES/
/home/root/workspace/EasyEdge-m1800-edgeboard/RES/1.jpg
```

3.5.3 Call HTTP Service

1. Load the driver once after the system starts-up (Or add system startup script)

```
insmod /home/root/workspace/driver/fpgadrv.ko
```

If the driver is not loaded, an error may be reported：

```
Failed to to fpga device: -1
```

Set system time (System time must be correct)

```
date --set "2019-5-18 20:48:00"
```

2. HTTP service function is included in the deployment package, which can be run directly

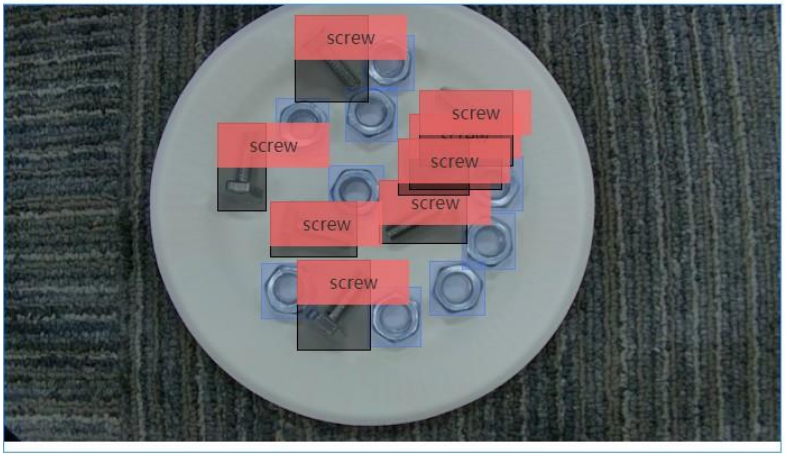
```
# ./easyedge_serving {RES directory} {serial number} {bounded host, default is 0.0.0.0}
{bounded port, default is 24401}
cd ${SDK_ROOT}
export LD_LIBRARY_PATH=./lib
./demo/easyedge_serving  /home/root/workspace/EasyEdge-m1800-edgeboard/RES/  "1111-1111-
1111-1111"
```

The start-up is successful if the log shows as below:

```
2019-07-18 13:27:05,941 INFO [EasyEdge] [http_server.cpp:136] 547974369280 Serving at
0.0.0.0:24401
```

Now you can directly input http://{fz5 IP address}:24401 in the browser to test the model effect in H5.

上传图片



结果

标签	置信度
nut	1.00
nut	1.00
nut	1.00
nut	0.99
screw	0.99
screw	0.98
screw	0.95
screw	0.92
screw	0.86
screw	0.85
screw	0.64
screw	0.34

3.5.4 HTTP private service request description

- http request parameters

Get parameter in URL：

parameter	Explanation	Defaults
threshold	Threshold filtering， 0~1	0.1

HTTP post body is the binary content of the image (No Base64, no JSON)

Request example for **Python**:

```
import requests


with open('./1.jpg', 'rb') as f:
    img = f.read()
    result =
        requests.post( 'http://127.0.0.1:
        24401/',
        params={'threshold': 0.1},
        data=img).json()
```

Request example for **JAVA**

- http data return

Field	Type	Other
error_code	Number	0 means success. If not, refer to message for specific error message
results	Array	The content is specific identification results
cost_ms	Number	Prediction time

Example of return data:


```
{
    "cost_ms": 52,
    "error_code": 0,
    "results": [
        {
            "confidence": 0.94482421875,
            "index": 1,
            "label": "IronMan",
            "x1": 0.059185408055782318,
            "x2": 0.18795496225357056,
            "y1": 0.14762254059314728,
            "y2": 0.52510076761245728
        },
        {
            "confidence": 0.94091796875,
            "index": 1,
            "label": "IronMan",
            "x1": 0.79151463508605957,
            "x2": 0.92310667037963867,
            "y1": 0.045728668570518494,
            "y2": 0.42920106649398804
        },
    }
```


Errors description

All active errors reported by the SDK are covered in the Edgestatus enumeration. At the same time, the SDK will show a detailed error log. Developers can open the debug log to check additional instructions:

```
global_controller()->log_config.enable_debug = true;
```

4.Advanced Guide

4.1 Develop Applications

4.1.1 Model acquisition

Currently Paddle-Mobile only supports Paddle trained models. If the models in your hands are different types of models, you need to perform model conversion before running. Verified networks include resnet, Inception, ssd, mobilenet, etc.

- **Training model**:
  If you don't have a model, you can use the model in sample or train the model by yourself.
  1. Train model through Paddlepaddle open source deep learning framework. Reference for detailed use: PaddlePaddle
  2. Through model through AI studio platform training model,
  Reference for detailed use: AI Studio
  3. You may upload labeled data on EasyDL platform to train the model. Reference for detailed use: EasyDL
- **Converting model**:
  1. If you already have a **Caffe** model, MYIR provides a corresponding conversion tool to help you convert it to a Paddle model. Reference for detailed use: X2Paddle_caffe2fluid
  2. If you already have a **TensorFlow** model, MYIR provides a corresponding conversion tool to help you convert it to a Paddle model. Reference for detailed use: X2Paddle_tensorflow2fluid

4.1.2 Connecting video data sources

FZ5 provides a variety of video input hardware interfaces and supports multiple protocols to input image data as data source. For example: HDMI, USB, GigE, etc.

1. **Video data input via USB protocol video**
   You can choose the UVC USB camera as the video source. Insert USB camera into USB interface of FZ5
2. **Video data input via HDMI protocol**
   You can select the HDMI video data output from HDMI video source. Connect the HDMI out interface of FZ5 through HDMI cable.
3. **Video data input via GigE protocol**
   You can choose the GIGE camera that supports the Linux system, and contact our company to adapt the official SDK of the camera. The hardware is connected to the FZ5 network port.

4.1.3 Load device driver

Using FZ5's acceleration function, the prediction library will calculate the op with a large amount of calculation through the driver to call FPGA to calculate. The driver needs to be loaded before running your own application. The compiled driver is in the directory /home/root/workspace/driver, providing two versions: the first version with no log output and the second version with log output.

Load driver

```
insmod /home/root/workspace/driver/fpgadrv.ko
```

Uninstall driver (In normal circumstances, you don't need to uninstall the driver. If you need to load the version with log output, you can uninstall it with the following command and then load

this version)

```
rmmod /home/root/workspace/driver/fpgadrv.ko
```

Set the driver to be automatically loaded

1） Add a self-starting script to the system

```
// Open the startup directory
cd /etc/init.d/
// Create a new startup script and edit it, the name can be customized
vim eb.sh
```

Script content

```
chmod +x /home/root/workspace/driver/fpgadrv.ko
insmod /home/root/workspace/driver/fpgadrv.ko
```

2 ）Establish soft links

```
cd /etc/rc5.d/
ln -s /etc/init.d/eb.sh S99eb
```

3 ）Change script permissions

```
chmod +x /etc/init.d/eb.sh
reboot
```

4.1.4 Using the prediction library

FZ5 supports Paddle-Moblie prediction library. The compiled prediction library is in /home/root/workspace/paddle-mobile.

How to use it: Copy the header file and dynamic library of the prediction library to your own application. You can also refer to the sample MYIR provides. Paddle-Moblie source code:

https://github.com/PaddlePaddle/paddle-mobile

4.1.5 Create Application

4.1.5.1 Add prediction library

Copy the dynamic libraries and header files in /home/root/workspace/paddle-mobilie/ to your project. Add a reference to the Paddle-Mobile library in CmakeLists.txt

```
set(PADDLE_LIB_DIR "${PROJECT_SOURCE_DIR}/lib" )
set(PADDLE_INCLUDE_DIR "${PROJECT_SOURCE_DIR}/include/paddle-mobile/" )

include_directories(${PADDLE_INCLUDE_DIR})
LINK_DIRECTORIES(${PADDLE_LIB_DIR})
...
target_link_libraries(${APP_NAME} paddle-mobile)
```

4.1.5.2 Add model

Copy your trained model to your project.

4.1.5.3 Add Prediction data sources

You can select pictures and camera data as the source of prediction data. To use the camera, you need to insert the corresponding camera.

- **USB Camera**

1 ）After plugging in the camera, check the device access through ls /dev/video*. Display as below means pass:

```
/dev/video0 /dev/video1 /dev/video2
```

/dev/video2 outputs YUV data for USB camera. When the application prompts that the device cannot be found, you can modify src/video_classify.cpp or src/video_detection.cpp. Check the camera connectivity through the video tool in /home/root/workspace/tools.

```
// src/video_classify.cpp line 169
config.dev_name = "/dev/video2";
```

2 ）The camera resolution can be modified

```
// src/video_classify.cpp line 170
config.width = 1280;
config.height = 720;
```

3 ）Run the video tool

```
// Read the USB camera data, collect a picture and save it to the local
   cd /home/root/workspace/tools/video
   ./v4l2demo -i /dev/video2 -j -n 1
// If in doubt, check the help
   ./v4l2demo -h
```

After executing the program, a .jpg file will be generated in the directory build, you can check whether the picture is correct or not. If no picture is generated, please check if the USB device is recognized.

- **HDMI video source**

  After inserting the HDMI video source, use the video tool in /home/root/workspace/tools to check the HDMI interface connectivity.

1） View the device, the directory is /dev/video1 in normal circumstances

```
ls /dev/video*
/dev/video0  /dev/video1
```

2） Set the video source parameters

```
media-ctl -v --set-format '"a0010000.v_tpg":0 [RBG24 1920x1080 field:none]'
```

3） Run the video tool

```
// Read HDMI video data, collect a picture and save to local
   cd /home/root/workspace/tools/video
   ./v4l2demo -i /dev/video1 -j -n 1
// If in doubt, check the help
   ./v4l2demo -h
```

After executing the program, a .jpg file will be generated in the directory build, you can check if the picture is correct. If no picture is generated, check whether the HDMI cable is connected correctly.

4.1.5.4 Call the prediction library to load the model and use the prediction data

- Initialize the model

```
 Predictor _predictor_handle = new Predictor();
 _predictor_handle->init(model, {batchNum, channel, input_height, input_width},
output_names);
```

- Prepare data
  1. Scale the picture to the specified size. If the neural network requires a fixed size, the picture needs to be scaled to that fixed size.
  2. Image preprocessing (Minus mean value, floating point conversion, normalization, etc.).
  3. Output data. Because FZ5 uses NHWC format, usually the data from the video is in NHWC format, so NHWC->NCHW conversion is not required.
- Predict data
  Call the predict interface of the API, transmit the processed data and obtain the prediction result

```
bool predict(const float* inputs, vector<float*> &outputs,vector<vector<int> >
&output_shapes);
```

4.2 Driver description

The driver module includes functions such as device management, memory management, IO setting, parameter management and command control, etc.

1. Device management

```
int open_device();
void close_device();
void reset_device();
```

2. Memory management

The driver will reserve its own dedicated memory from the system memory for itself and the FPGA device. This part of memory cannot be seen or cannot be operated independently by Linux OS without the driver. The driver is needed to perform the corresponding memory management (allocate, release, map, copy, etc.)

```
void* fpga_malloc(size_t size);
void fpga_free(void* ptr);
void fpga_copy(void* dst, const void* src, size_t num);
int fpga_flush(void* address, size_t size);
int fpga_invalidate(void* address, size_t size);
```

3. IO setting, parameter management and command control

At the same time, the driver also provides functions such as IO setting, parameter management, command control for the upper layer (The top-layer main control application) and completes the corresponding communication settings with the FPGA device.

```
int PerformBypass(const struct BypassArgs& args);
int ComputeFpgaConv(const struct ConvArgs& args);
int ComputeFpgaPool(const struct PoolingArgs& args);
int ComputeFpgaEWAdd(const struct EWAddArgs& args);
int ComputeFPGAConcat(const struct ConcatArgs& args);
int ComputeScale(const struct ScaleArgs& args);
int ComputeNormalize(const struct NormalizeArgs& args);
int ConfigPowerParameter(const struct PowerParameterArgs& args);
int ConfigNormalizeParameter(const struct NormalizeParameterArgs& args);
int ConfigInplace(const struct InplaceArgs& args);
```

4.3 Prediction library description

1. Paddle-Mobile

Paddle-Mobile is a project under PaddlePaddle structure and is a deep learning prediction framework dedicated to embedded platforms. FZ5 uses the Paddle-Mobile prediction library implemented by FPGA under Paddle-Mobile.

2. NHWC

Based on the characteristics of FPGA, Paddle-Mobile's FPGA implementation data format is NHWC. Please pay attention to this point when developing your own application or modifying the Paddle-Mobile code.

```
void convert_to_hwc(float **data_in, int channel, int height, int width)
```

3. FP16 (Implemented as half)

The OP and Tensor data realized by FPGA are FP16. When encountering OP realized by CPU, FP16 needs to be converted to FP32 (i.e. Float). On the contrary, FP32 needs to be converted to FP16.

```
// FP16->FP32
float* float_data = (float*)fpga::fpga_malloc(height * cw_alinged * sizeof(float));
fpga::to_float(const_cast<float*>(input_half->data<float>()), float_data, height *
cw_alinged);

// FP32->FP16
fpga::to_half(output_boxes_dataptr, boxes_data, output_boxes->numel());
```

4. Align

Based on the characteristics of FPGA, before using OP realized by FPGA, the data of Tensor needs to be aligned based on C*W by 16-bit multiple. On the contrary, the previous node is implemented by FPGA, and the next node needs to be aligned when implemented by CPU.

```
void align_element(char** data_in, int num, int chw);
void align_num(char** data_in, int num_per_div_before_alignment, int num,int chw);
```

5. OP customization

When Paddle-Mobile's existing OP cannot meet your model needs, you can add or customize an OP. For details, please refer to Paddle-Mobile OP code design.

Appendix 1 Warranty & Technical Support Services

MYIR Tech Limited (MYIR for short) is a global provider of ARM hardware and software tools, design solutions for embedded applications. We support our customers in a wide range of services to accelerate your time to market.

MYIR is an ARM Connected Community Member and work closely with ARM and many semiconductor vendors. We sell products ranging from board level products such as development boards, single board computers and CPU modules to help with your evaluation, prototype, and system integration or creating your own applications. Our products are used widely in industrial control, medical devices, consumer electronic, telecommunication systems, Human Machine Interface (HMI) and more other embedded applications. MYIR has an experienced team and provides custom design services based on ARM processors to help customers make your idea a reality.

The contents below introduce to customers the warranty and technical support services provided by MYIR as well as the matters needing attention in using MYIR's products.

Service Guarantee
MYIR regards the product quality as the life of an enterprise. We strictly check and control the core board design, the procurement of components, production control, product testing, packaging, shipping and other aspects and strive to provide products with best quality to customers. We believe that only quality products and excellent services can ensure the long-term cooperation and mutual benefit.

Price
MYIR insists on providing customers with the most valuable products. We do not pursue excess profits which we think only for short-time cooperation. Instead, we hope to establish long-term cooperation and win-win business with customers. So we will offer reasonable prices in the hope of making the business greater with the customers together hand in hand.

NCNR (Non-cancellation of orders and non-return of goods)
No returns or cancellations will be accepted without prior written agreement from MYIR.

Delivery Time
MYIR will always keep a certain stock for its regular products. If your order quantity is less than the amount of inventory, the delivery time would be within three days; if your order quantity is greater than the number of inventory, the delivery time would be always four to six weeks. If for any urgent delivery, we can negotiate with customer and try to supply the goods in advance.

Technical Support
MYIR has a professional technical support team. Customer can contact us by email (support@myirtech.com), we will try to reply you within 48 hours. For mass production and customized products, we will specify person to follow the case and ensure the smooth production.

After-sale Service

MYIR offers one year free technical support and after-sales maintenance service from the purchase date. The service covers:

1. Technical support service

a) MYIR offers technical support for the hardware and software materials which have provided to customers;
b) To help customers compile and run the source code we offer;
c) To help customers solve problems occurred during operations if users follow the user manual documents;
d) To judge whether the failure exists;
e) To provide free software upgrading service.

However, the following situations are not included in the scope of our free technical support service:

a) Hardware or software problems occurred during customers' own development;
b) Problems occurred when customers compile or run the OS which is tailored by themselves;
c) Problems occurred during customers' own applications development;
d) Problems occurred during the modification of MYIR's software source code.

2. After-sales maintenance service

The products except LCD, which are not used properly, will take the twelve months free maintenance service since the purchase date. But following situations are not included in the scope of our free maintenance service:

a) The warranty period is expired;
b) The customer cannot provide proof-of-purchase or the product has no serial number;
c) The customer has not followed the instruction of the manual which has caused the damage the product;
d) Due to the natural disasters (unexpected matters), or natural attrition of the components, or unexpected matters leads the defects of appearance/function;
e) Due to the power supply, bump, leaking of the roof, pets, moist, impurities into the boards, all those reasons which have caused the damage of the products or defects of appearance;
f) Due to unauthorized weld or dismantle parts or repair the products which has caused the damage of the products or defects of appearance;
g) Due to unauthorized installation of the software, system or incorrect configuration or computer virus which has caused the damage of products.

Warm tips:

1) MYIR does not supply maintenance service to LCD. We suggest the customer first check the LCD when receiving the goods. In case the LCD cannot run or no display, customer should contact MYIR within 7 business days from the moment get the goods.
2) Please do not use finger nails or hard sharp object to touch the surface of the LCD.
3) MYIR suggests user purchasing a piece of special wiper to wipe the LCD after long time use, please avoid clean the surface with fingers or hands to leave fingerprint.
4) Do not clean the surface of the screen with chemicals.
5) Please read through the product user manual before you using MYIR's products.
6) For any maintenance service, customers should communicate with MYIR to confirm the issue first. MYIR's support team will judge the failure to see if the goods need to be returned for repair service, we will issue you RMA number for return maintenance service after confirmation.

3. Maintenance period and charges
a) MYIR will test the products within three days after receipt of the returned goods and inform customer the testing result. Then we will arrange shipment within one week for the repaired goods to the customer. For any special failure, we will negotiate with customers to confirm the maintenance period.
b) For products within warranty period and caused by quality problem, MYIR offers free maintenance service; for products within warranty period but out of free maintenance service scope, MYIR provides maintenance service but shall charge some basic material cost; for products out of warranty period, MYIR provides maintenance service but shall charge some basic material cost and handling fee.

4. Shipping cost
During the warranty period, the shipping cost which delivered to MYIR should be responsible by user; MYIR will pay for the return shipping cost to users when the product is repaired. If the warranty period is expired, all the shipping cost will be responsible by users.

5. Products Life Cycle
MYIR will always select mainstream chips for our design, thus to ensure at least ten years continuous supply; if meeting some main chip stopping production, we will inform customers in time and assist customers with products updating and upgrading.

Value-added Services

1. MYIR provides services of driver development base on MYIR's products, like serial port, USB, Ethernet, LCD, etc.
2. MYIR provides the services of OS porting, BSP drivers' development, API software development, etc.
3. MYIR provides other products supporting services like power adapter, LCD panel, etc.
4. ODM/OEM services.

Contact Us

Support Email: support@myirtech.com
Sales Email: sales@myirtech.com
Phone: +86-755-22984836
Fax: +86-755-25532724