

百度大脑EdgeBoard边缘AI计算盒及计算卡(FZ5) 使用手册 V1.0

版本记录

版本号	说明	时间
V1.0	初始版本	2020/08/03

目录

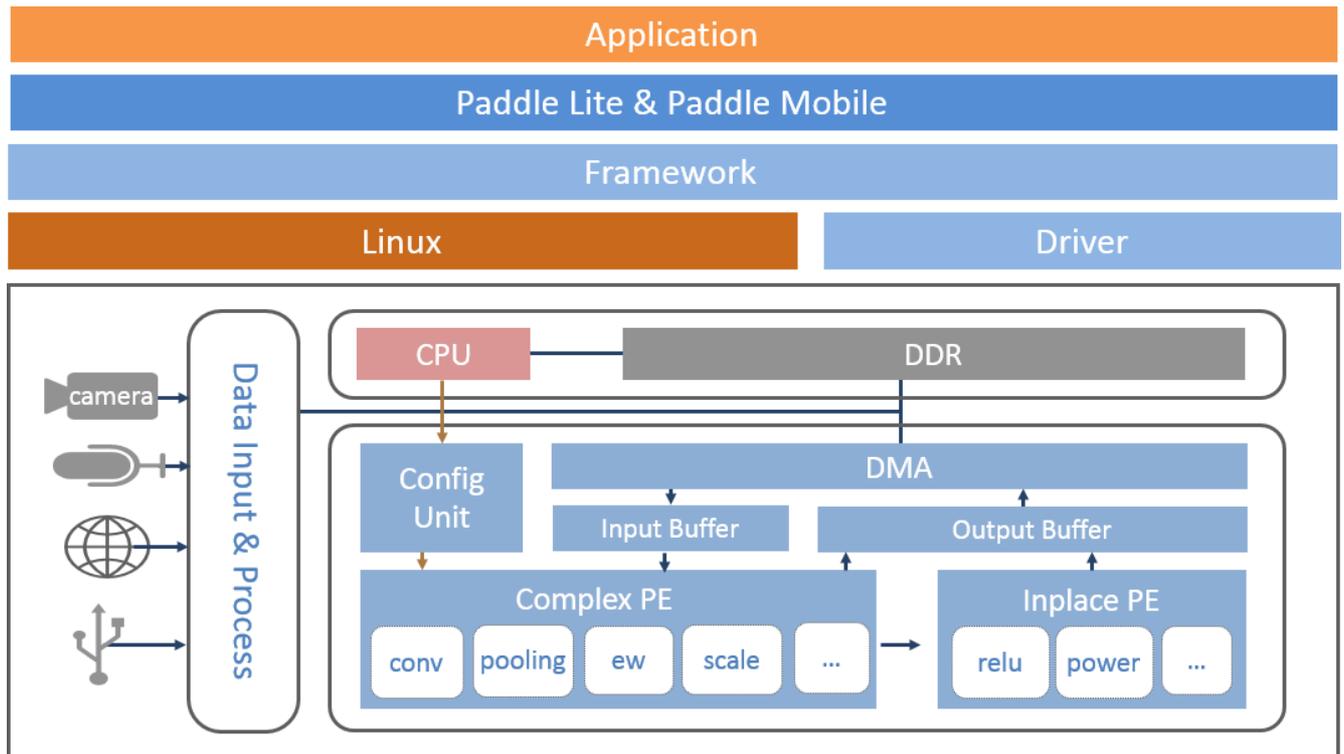
1.软核简介	5
1.1 软件介绍.....	5
1.2 常用模型（未量化裁剪）在FZ5上的性能数据.....	5
2.启动与连接.....	6
2.1 启动准备.....	6
2.2 连接方式一：SSH连接	6
2.2.1 Windows使用SSH连接网口方法.....	6
2.2.2 MAC使用SSH网络服务连接网口方法.....	7
2.3 连接方式二：串口连接（备选）	8
2.3.1 Windows使用SecureCRT连接串口方法	8
2.3.2 MAC使用minicom连接串口方法	10
3.调试设备	10
3.1 更改设备网络配置方法.....	10
3.2 文件拷贝.....	11
3.2.1 通过FTP实现文件拷贝（适用于windows系统）	11
3.2.2 通过samba协议实现文件拷贝（适用于Mac OS系统）	12
3.3 系统目录介绍	12
3.4.1 分类模型示例	12
3.4.2 目标检测示例	14
3.4.3 推理结果输出显示.....	16
3.4.4 Video视频输入方式.....	18
3.5 运行EasyDL平台模型预测示例.....	18
3.5.1 EasyDL的使用方式.....	18
3.5.2 运行SDK.....	23
3.5.3 HTTP服务调用.....	24
3.5.4 HTTP 私有服务请求说明	25
4.进阶指南	26
4.1 开发应用.....	27
4.1.1 模型获得.....	27
4.1.2 连接视频数据源	27
4.1.3 加载驱动.....	27
4.1.4 使用预测库.....	28
4.1.5 创建应用.....	28
4.2 驱动说明.....	30
4.3 预测库说明.....	31
附录一售后服务与技术支持	32

前言

百度大脑EdgeBoard计算盒/计算卡(FZ5)是一款基于Xilinx Zynq Ultrascale CZU5EV芯片打造的深度学习计算盒/计算卡，芯片内部集成了4核ARM A53处理器+GPU+FPGA的架构，具有多核心处理能力、FPGA可编程能力以及视频流硬件解码能力等特点；内置了基于Linux操作系统+百度深度学习平台-飞桨（Paddle）定制的深度学习软核，深度兼容百度大脑模型资源与工具平台（EasyDL/AIStudio），可高效、快速的实现模型的训练-部署-推理等一系列流程，极大降低了开发验证、产品集成、科研教学、项目部署的门槛。

1. 软核简介

1.1 软件介绍



FZ5 计算卡内部搭载Linux系统，开发者可以基于Linux系统进行应用程序进行开发。（主要调用流程：1.应用程序获取视频输入->2.调用预测库加载模型->3.调度模型和底层驱动加速模块进行计算->4.获得运行结果）

1.2 常用模型（未量化裁剪）在FZ5上的性能数据

网络	输入尺寸	单帧耗时
resnet50	224 x 224	42ms
mobilenet-v1	224 x 224	10ms
inception-v2	299 x 299	41ms
inception-v3	299 x 299	70ms
resnext	224x224	69ms
mobilenet-ssd	300 x 300	24ms
mobilenet-ssd-640	640 x 640	79ms
vgg-ssd	300x300	246ms
yolov3	608x608	582ms

注：软核仍在持续升级，性能将同步提升。相同网络结构不同版本对算力要求不同，如有具体项目应用，可联系官方团队申请定制优化。

2.启动与连接

2.1 启动准备

- 1.系统默认是从TF卡启动，上电前，需要将配套的TF卡插到设备的TF卡座中，才能正常使用设备的功能；
- 2.设备调试支持串口调试和网口调试，串口调试对应设备的控制台，在进行一般调试时可能会有冗余的打印信息，所以建议用户使用网口调试，设备在出厂时设置了静态ip: 192.168.1.254，用户可通过网线直连电脑或连接路由设备（网线需用户自行准备），使用SSH协议登录系统，具体使用方法详见后文。
- 3.无论是串口调试，还是网口调试，设备在启动后都要求用户输入login和密码，即root & root，输入成功，即可登录设备系统，系统中有自带的深度学习预装环境以及为模型推理的sample，用户在下面的使用中会有详细介绍

2.2 连接方式一：SSH连接

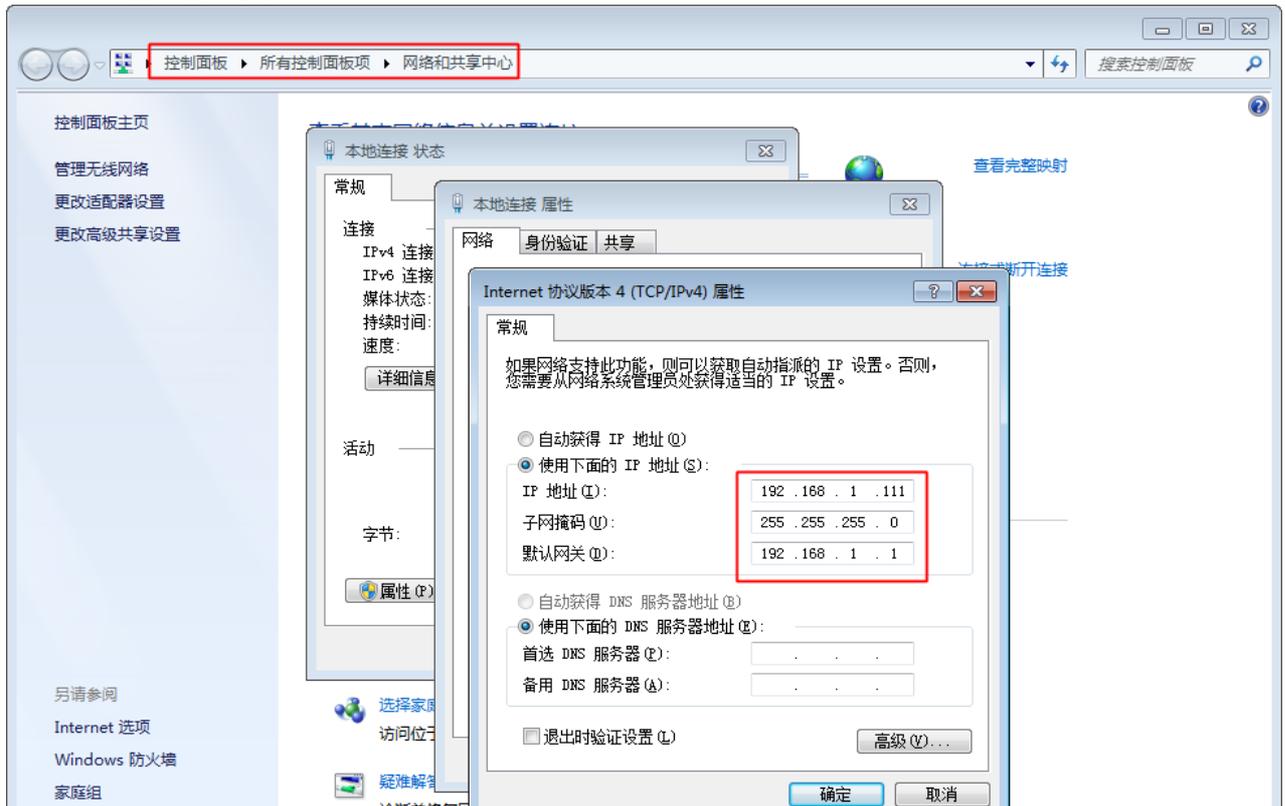
- FZ5 出厂默认参数为静态ip=192.168.1.254，netmask=255:255:255:0，gateway=192.168.1.1 硬件连接方
- 法：使用网线一端连接FZ5，另一端连接host电脑或者路由器，设置电脑或路由器ip和FZ5在同一网段，即可使用SSH登录

具体步骤如下：

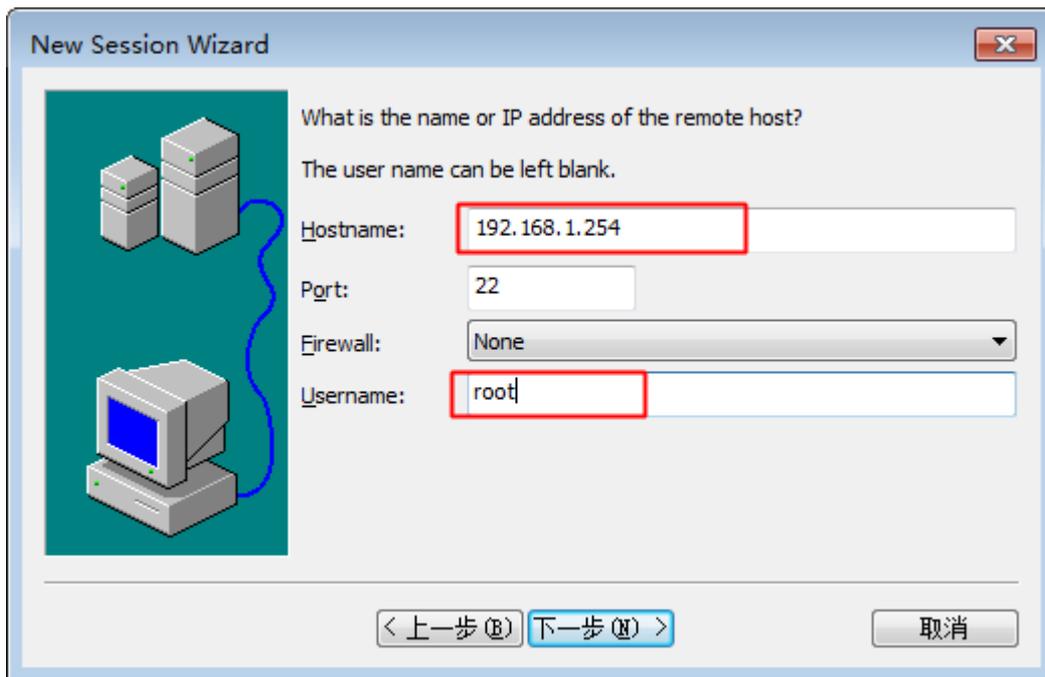
2.2.1 Windows使用SSH连接网口方法

1. 安装调试工具，推荐SecureCRT工具（可百度搜索安装）；

2. 配置电脑或路由器ip和设备在同一网段下,当电脑和设备直连时需要手动设置电脑ip, 打开网络和共享中心-->本地连接-->属性-->Internet协议版本4, 手动配置ip地址: 192.168.1.111, 子网掩码: 255.255.255.0, 默认网关: 192.168.1.1,如下图所示。



3. 在secure CRT中新建窗口, connect-->New Session-->Protocol选择SSH2,点击下一步, Hostname为FZ5的ip, 出厂默认192.168.1.254, port=22, 下一步, 完成, 在弹窗中输入username=root, password=root, 即可进入系统。



2.2.2 MAC使用SSH网络服务连接网口方法

1. 配置电脑（或者路由器）ip为192.168.1.xxx（1<xxx<253），保证电脑和FZ5的ip在同一网段 配
置步骤：系统偏好设置-->网络-->高级-->TCP/IP。

IPv4配置示例：手动，IPv4地址：192.168.1.111，子网掩码：255.255.255.0，路由器：192.168.1.1



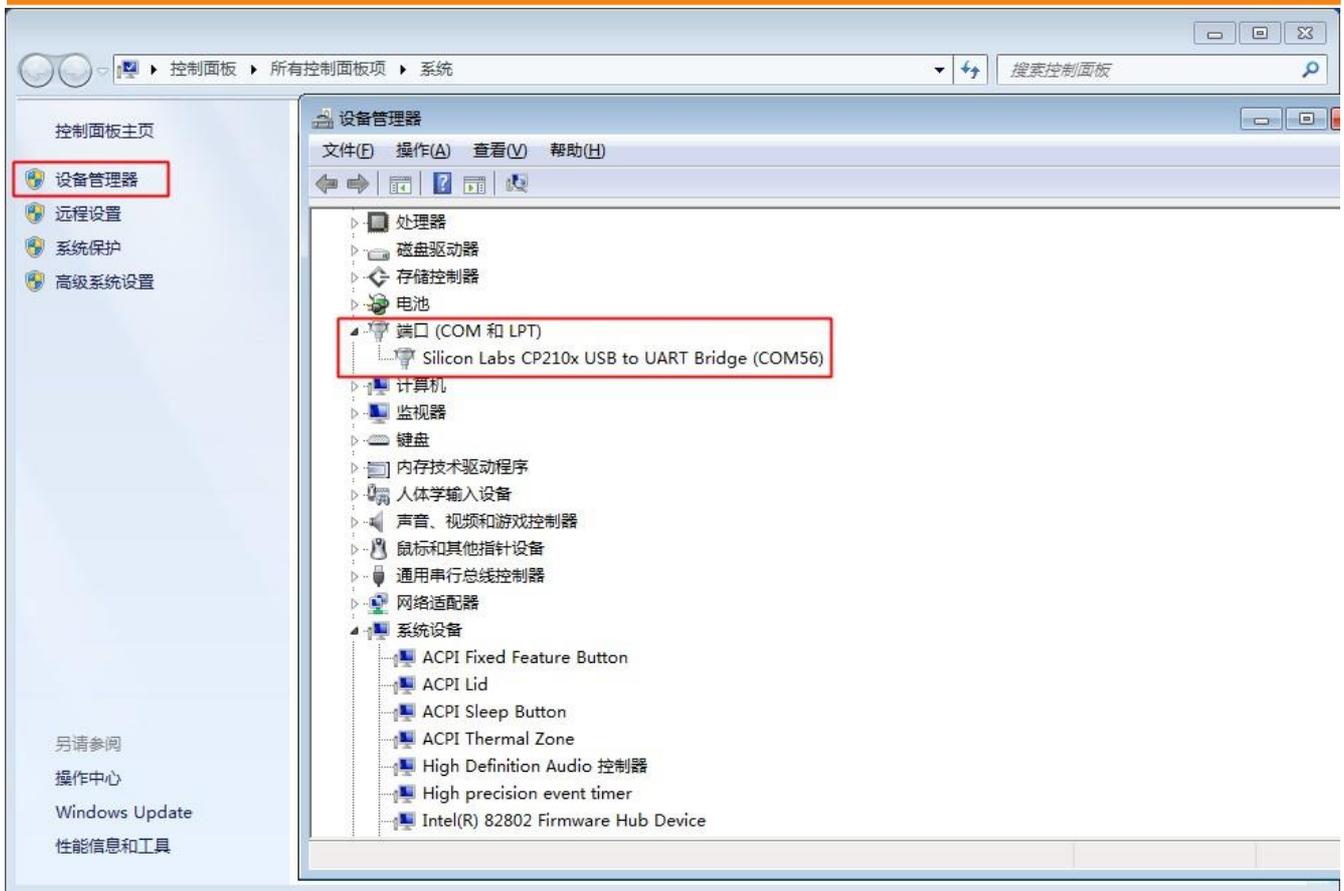
2. **打开Terminal:** Launchpad->其它(文件夹)->终端（即Terminal程序）
3. 在Terminal中输入ssh root@ip（FZ5的IP地址），默认地址为 ssh root@192.168.1.254，然后再输入 login&password为root&root,即可登录FZ5系统

2.3连接方式二：串口连接（备选）

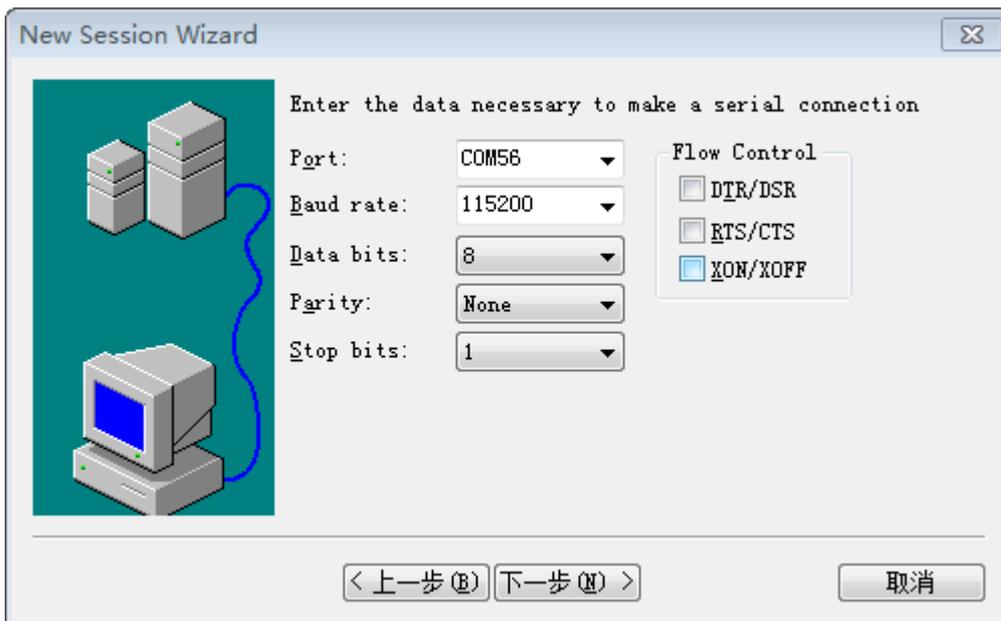
如果出现ssh连接不上，或者（设备ip动态获取后）需要查看ip，需要使用串口进入设备的控制台
可使用micro usb数据线一端连接FZ5的USB UART接口，即usb转串口，一端连接电脑。

2.3.1 Windows使用SecureCRT连接串口方法

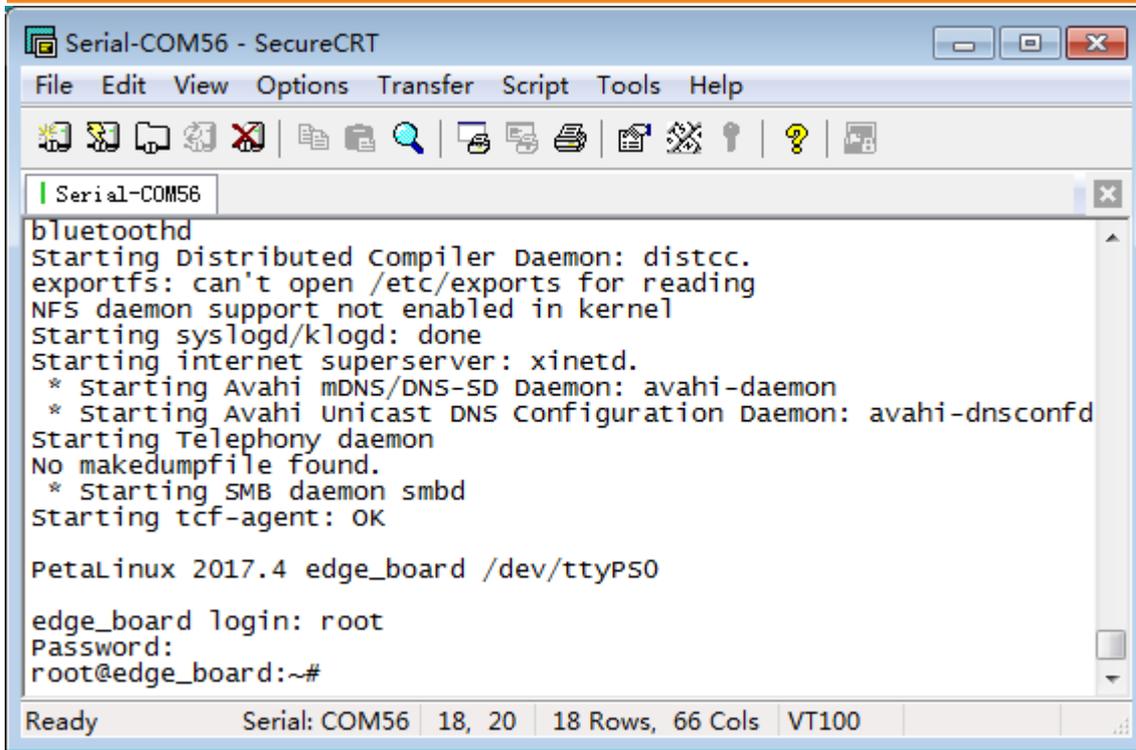
- 1.安装SecureCRT软件和串口驱动 CP210x_Windows_Drivers（初次使用需安装驱动，安装包可百度搜索）
- 2.保证电脑已连接FZ5的USB UART接口，【右击“我的电脑”->属性->设备管理器】查看设备管理器中映射的端口号，如图示，端口号为COM56



3.打开SecureCRT，新建窗口connect->New Session->Protocol选择Serial，波特率选择115200，Flow Control不选，如下图所示



4.点击【完成】 - 【Connect】按钮，SecureCRT会连接到计算卡上的串口，上电后即可看到启动信息，待启动完成后输入用户名和密码root/root,即可进入设备系统。如下图所示



2.3.2 MAC使用minicom连接串口方法

- 1.在电脑中安装串口驱动：SiLabsUSBDriverDisk.dmg（初次使用需安装，可百度搜索安装方法）
- 2.mac安装minicom工具（可百度搜索安装方法）
- 3.打开Terminal: Launchpad->其它(文件夹)->终端（即Terminal程序），在terminal终端输入minicom -s 进行配置
- 4.配置内容如下，配置完成后，连接上EdgeBoard，在terminal终端输入minicom即可。

- 选择Serial port setup，配置如下：
 - 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.调试设备

3.1 更改设备网络配置方法

设备出厂默认为静态ip地址192.168.1.254，如果多个设备同时连到同一个局域网，则需要更改设备为不同的ip地址，或者改为动态获取ip的方式，网络配置文件路径为/etc/network/interfaces

```
//打开并编辑interfaces文件
vim /etc/network/interfaces
```

静态ip配置

```
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
```

动态ip配置

```
auto lo
iface lo inet loopback

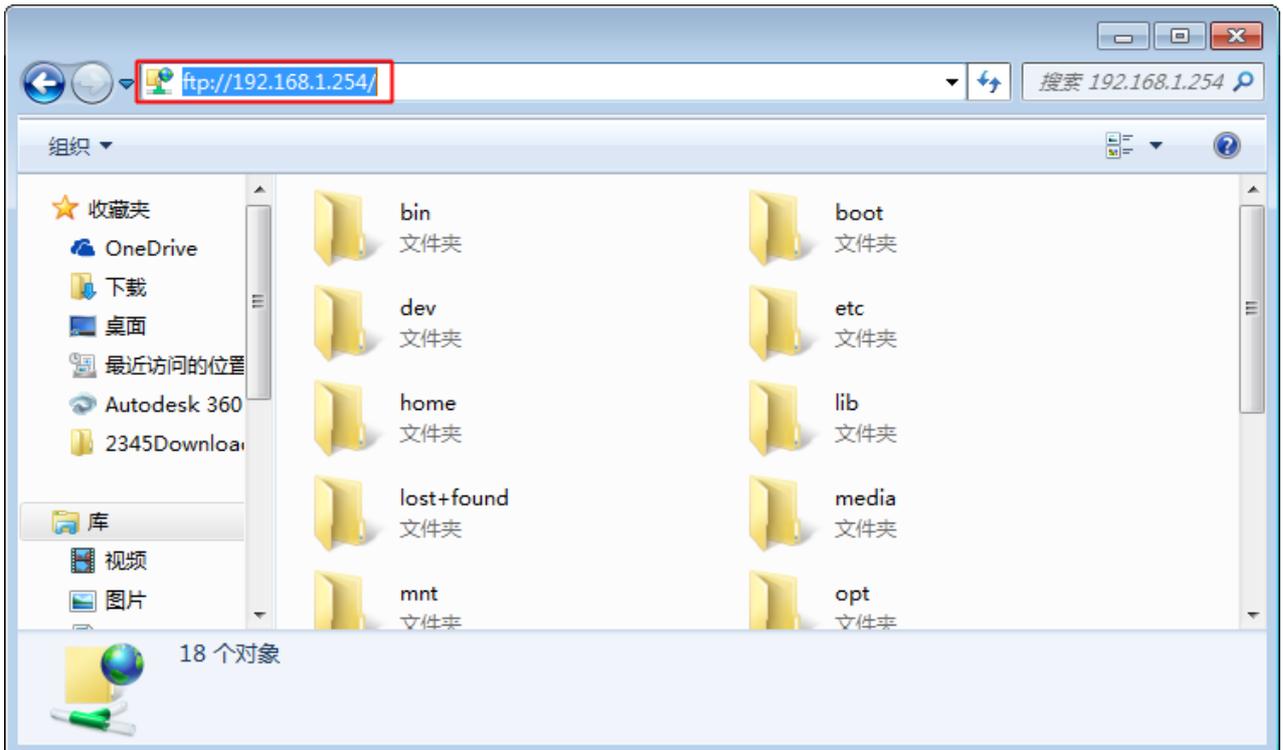
auto eth0
iface eth0 inet dhcp
```

3.2 文件拷贝

FZ5支持ssh、samba、ftp等网络协议，可轻松通过网络进行数据通讯以及文件拷贝的工作，使用这一功能在 后面软件升级、用户定制时都会有广泛的应用。

3.2.1 通过FTP实现文件拷贝（适用于windows系统）

1. 【Windows+R快捷键组合，输入ipconfig】ifconfig命令查看设备ip，保证设备ip和windows电脑的ip在同一网段，在文件夹输入框里直接输入ftp://192.168.1.254。根据提示输入用户名root，密码root。即可进入设备系统



2. 打开home-->root-->workspace目录，workspace是root用户下应用程序所在的目录，直接拷贝文件到workspace，或者从workspace中拷贝文件到电脑即可。

3.2.2 通过samba协议实现文件拷贝（适用于Mac OS系统）

1. 配置并保证设备ip和MAC的ip在同一网段（参见上文：MAC使用SSH网络服务连接网口方法）
2. 配置完成后，点击Finder-->前往-->连接服务器，输入smb://ip，例如smb://192.168.1.254,用户名root，密码root。
3. finder中出现设备的文件目录，打开home-->root-->workspace目录，workspace是root用户下应用程序所在的目录，可以直接通复制粘贴命令进行电脑和设备间文件的拷贝。

3.3 系统目录介绍

内容	目录	备注
paddle-mobile	/home/root/workspace/paddle-mobile	paddle-mobile预测库
driver	/home/root/workspace/driver	驱动文件
sample	/home/root/workspace/sample	示例代码
tools	/home/root/workspace/tools	调试工具

3.4 运行Sample

当您连接上FZ5后，可以运行我们提供的深度学习示例，位于home/root/workspace/sample。

示例	说明
classification	分类模型示例
Detection	目标检测模型示例

3.4.1 分类模型示例

读取一张本地图片，调用模型进行推理，并输出结果。

考虑到简单通用性，该示例，从json文件中读取模型和图片信息，加载并执行。执行时需要指定 相应的配置文件。如

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

工程目录结构:

```
├─ CMakeLists.txt // cmake 工程配置文件。
├─ include
│   └─ io // paddle_mobile 头文件目录
│       ├── paddle_inference_api.h
│       ├── type_define.h
│       └─ types.h
├─ configs // 配置文件目录
│   └─ Inceptionv2
│       └─ zebra.json 斑马图片的配置
```

```

|   |— Inceptionv3
|   |   └─ zebra.json 斑马图片的配置
|   |— mobilenetv1
|   |   └─ zebra.json 斑马图片的配置
|   └─ resnet50
|       └─ drink.json 怡宝图片的配置
|— lib
|   └─ libpaddle-mobile.so
|— models // 模型文件目录
|   |— Inceptionv2
|   |— Inceptionv3
|   |— mobilenetv1
|   └─ resnet50
|— src
|   |— json.hpp // json 解析库
|   |— video_classify.cpp // 视频推理示例
|   └─ image_classify.cpp // 图片推理示例
└─ README.md
    
```

下面是配置文件示例。

```

{
  "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	模型目录存放的位置
combined_model	是否为融合模型，只有两个文件的是融合模型
input_width	输入网络的图片尺寸 输入图片会缩放到该大小
input_height	输入网络的图片尺寸
Image	进行分类的图片输入
Mean	平均值
Scale	输入网络前预算处理为 $(x - \text{mean}) * \text{scale}$
Format	网络所需要的格式，OpenCV默认是BGR

其它分类网络也可以通过添加/修改 配置文件实现，无须修改代码。

示例操作步骤：

1.加载驱动,系统启动后加载一次即可（出厂已默认自动加载）

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

2.编译示例，FZ5上具有编译能力，进入到sample/classification示例的build目录下进行编译

```
// 打开示例目录
cd /home/root/workspace/sample/classification
// 如果没有build目录，创建一个
mkdir build
cd build
rm -rf *
// 调用cmake 创建 Makefile
cmake ..
// 编译工程。
make
```

编译结束后会在build 目录生成如下几个文件。

image_classify 读取本地图片推理示例。

video_classify 读取摄像头数据进行推理，要连接摄像头(USB Camera)才能使用。如需显示结果还要连接DP显示器。

针对一些需要将HDMI作为视频源输入接口的工业应用场景，我们提供了一路HDMI IN接口，系统以及软硬件均做了相关适配，可以直接使用。

3.执行示例

```
使用图片推理
./image_classify ../configs/resnet50/drink.json

使用视频推理，必须保证USB摄像头和显示器同时连接才可正确运行
startx //打开桌面环境
./video_classify ../configs/resnet50/drink.json
```

3.4.2 目标检测示例

和分类不同，物体检测除了能知道物体的类型，还能检测出物体所在的位置坐标。物体检测也分了两个示例，一个是在图片上检测物体，并绘制出坐标信息。还有通过摄像头采集视频，检测在屏幕上绘制坐标信息。

工程目录结构:

```
├─ CMakeLists.txt // cmake 工程配置文件。
├─ include
│   └─ io // paddle_mobile 头文件目录
│       └─ paddle_inference_api.h
│       └─ type_define.h
│       └─ types.h
```

```

├─ configs // 配置文件目录
│   └─ mobilenet-ssd
│       └─ screw.json 螺丝螺母图片的配置
│   └─ mobilenet-ssd-640
│       └─ screw.json 螺丝螺母图片的配置
│   └─ vgg-ssd
│       └─ screw.json 螺丝螺母图片的配置
├─ lib
│   └─ libpaddle-mobile.so
├─ models // 模型文件目录
│   └─ mobilenet-ssd
│   └─ mobilenet-ssd-640
│   └─ vgg-ssd
├─ src
│   └─ json.hpp // json 解析库
│   └─ video_detection.cpp // 视频推理示例
│   └─ image_detection.cpp // 图片推理示例
└─ README.md
    
```

下面是配置文件示例。

```

{
  "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	模型目录存放的位置
combined_model	是否为融合模型，只有两个文件的是融合模型
input_width	输入网络的图片尺寸 输入图片会缩放到该大小
input_height	输入网络的图片尺寸
Image	进行分类的图片输入
Mean	平均值
Scale	输入网络前预算处理为 $(x - \text{mean}) * \text{scale}$
Format	网络所需要的格式，OpenCV默认是BGR
Threshold	信心值阈值

其它分类网络也可以通过添加/修改 配置文件实现，无须修改代码。

示例操作步骤:

1.加载驱动,系统启动后加载一次即可(出厂已默认自动加载)

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

2.编译示例, FZ5上具有编译能力,进入到sample/detection示例的build目录下进行编译

```
cd /home/root/workspace/sample/detection
// 如果没有build目录, 创建一个
mkdir build
cd build
rm -rf *
// 调用cmake 创建 Makefile
cmake ..
// 编译工程。
make
```

编译结束后会在build 目录生成如下几个文件。

image_classify 读取本地图片推理示例。

video_classify 读取摄像头数据进行推理,要连接USB摄像头或者HDMI视频源才能使用。如需显示结果需要连接DP显示器或者HDMI显示器或者VGA显示器,使用方法参见2.2.3

3.执行示例

```
使用图片推理
./image_detection ../configs/vgg-ssd/screw.json

使用视频推理, 必须保证USB摄像头和显示器同时连接才可正确运行
startx //打开桌面环境
./video_detection ../configs/vgg-ssd/screw.json
```

FZ5硬件平台提供的是mini DP(DisplayPort)输出接口,可以用来连接支持DP口的显示器或者连接HDMI接口的显示器或者VGA接口的显示器。

3.4.3 推理结果输出显示

1. miniDP接口显示器: 使用公对公的mini DP转DP线连接FZ5和DP显示器即可。



2. HDMI接口显示器：需要使用【注意：主动式miniDP公转HDMI母】转换线来连接FZ5和HDMI显示器。[参考链接1](#)



3. VGA显示器：使用miniDP公转VGA母转接线连接FZ5和VGA显示器即可。[参考链接](#)



FZ5提供的软件系统支持一个精简版的Linux桌面环境，可以用来实时显示程序运行的效果。

在上电使用前请确保显示器和FZ5已经通过miniDP连接线连接好，进入系统后默认是进入终端命令行的环境。可以通过以下命令进入和退出桌面环境

```
startx //打开桌面环境  
stopx //关闭桌面环境
```

对于我们提供的演示示例，本身通过opencv已经支持将预测结果可视化的通过桌面窗口显示出来，运行方法参考3.4.1和3.4.2。

3.4.4 Video视频输入方式

FZ5支持USB、GIGE、HDMI 等协议的视频数据输入，可以作为各种场景的视频流的处理模块。

针对一些需要将HDMI作为视频源输入接口的应用场景，我们提供了一路HDMI IN接口，系统以及软硬件均做了相关适配，可以直接使用。

3.5 运行EasyDL平台模型预测示例

3.5.1 EasyDL的使用方式

EasyDL是一站式的深度学习模型训练和服务平台，提供可视化的操作界面，只需上传少量图片就可以获得高精度模型，具体可以参考[EasyDL官网](#)，通过EasyDL进行数据训练步骤如下：

3.5.1.1 选择训练类别

根据通用场景可以选择“图像分类”或者“物体检测”。

EasyDL产品

通用场景产品



图像分类

定制识别一张图中是否是某类物体/状态/场景，适合图片中主体或者状态单一的场景。



物体检测

定制识别图中每个物体的位置、名称。适合有多个主体、或要识别位置及数量的场景。

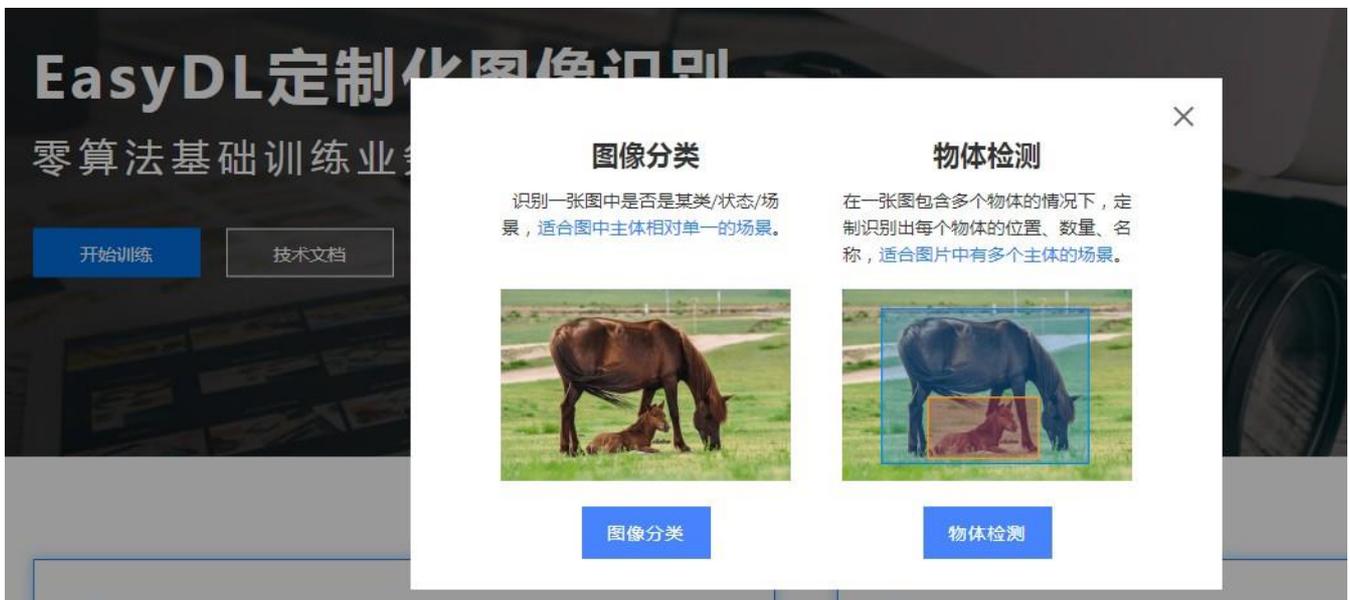


声音分类

定制识别当前音频是什么类型的声音。多应用于生产或泛安防场景中监控异常声音等。

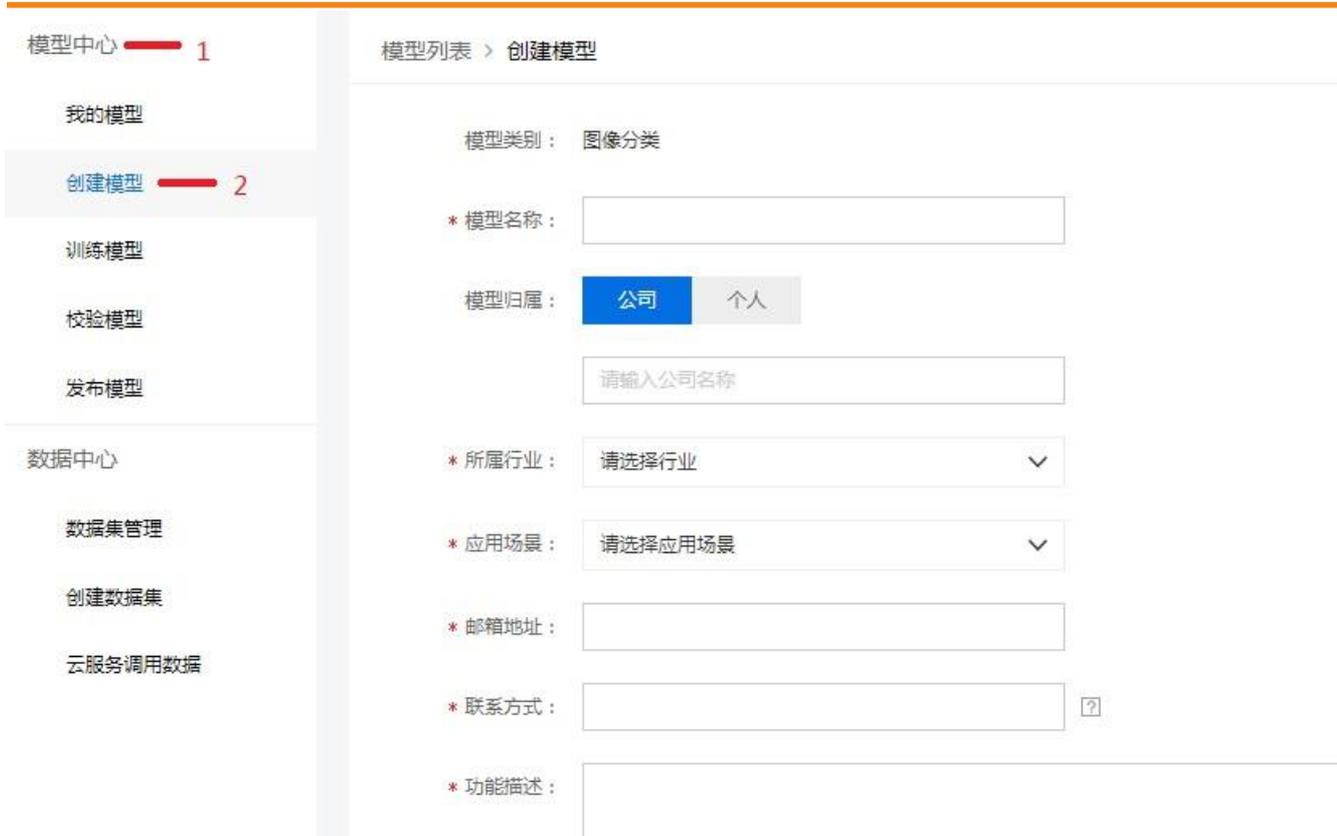
3.5.1.2 “图像分类”训练

选择“图像分类”，进入训练界面，选择“开始训练”，弹出图像分类和物体检测的对话框，选择需要的模型类型，如选择图像分类。



3.5.1.3 创建模型

进入模型中心，选择“创建模型”，根据个人需求填写模型的内容，带*号为必填选项。填写ok后选择“下一步”



3.5.1.4 训练模型

进入“训练模型”，选择模型类别，算法选择“通用”，训练方式选“默认”，因需要离线部署在设备上，**必须勾选“离线SDK”**，选择“添加训练数据”完成之后，勾选“增加识别结果为其他的默认分类”，选择“开始训练”。根据训练数据集的大小训练的时间会有些差异，一般100张左右的图片最快10分钟可训练完毕。





3.5.1.5 生成SDK

1. 训练完成后，在我的模型中申请发布模型，选择软硬一体方案中的"EdgeBoard + 专用SDK"，提交申请，等待审核



2. 模型审核成功后，在“我的模型”处，点击“服务详情”按钮，在弹出的对话框中，选择下载SDK。



3、获取序列号

点击“管理序列号”跳转至百度云-->EasyDL定制训练平台-->离线SDK管理界面，查看用于激活sdk的序列号。

序列号管理

按单台设备激活

按产品线激活

使用说明

1. 按单台设备获得授权激活SDK（如需开发APP，建议选择按产品线激活），操作步骤：

① 下载所需模型已发布的离线SDK → ② 在下方新增序列号 → ③ 设备端部署离线SDK，并联网激活 → ④ 离线使用

2. 每发布一个模型即可新增2个序列号，联网激活后有效期为3个月，可申请延期

使用过程中有问题可以[提交工单](#)或咨询EasyDL官方QQ群（群号：185289236）

+ 新增序列号

设备名	激活状态	序列号	激活时间
自定义设备	未激活		/

3.5.1.6 在FZ5里安装SDK

1. 下载的软件部署包包含了简单易用的SDK和Demo。只需简单的几个步骤，即可快速部署运行FZ5。部署包文件结构如下

```

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. 使用序列号License 激活SDK

打开demo.cpp文件

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

写入license序列号

将set_licence_key函数中的字符串参数"set your license here"替换为序列号License即可

```
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. 将SDK放到FZ5系统/home/root/workspace/目录下（放入方法参见上文“文件拷贝”）进行解压，然后按下述方法进行启动运行。

3.5.2 运行SDK

- 1.加载驱动,系统启动后加载一次即可（也可以加系统启动脚本，具体参见4.1.3）

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

若未加载驱动，可能报下面的错误：

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

设置系统时间（系统时间必须正确）

```
date --set "2020-7-28 20:48:00"
```

- 2.编译

```
//进入cpp文件的demo文件夹

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

// 如果没有build目录, 创建一个
mkdir build
cd build
rm -rf *
//调用cmake 创建 Makefile
cmake ..
// 编译工程
make
```

3.执行示例

```
//在build目录下运行执行文件
./easyedge_demo {RES资源文件夹路径} {测试图片路径}
```

例：在SDK中放入需要预测的图片，如将预测图片放入RES文件夹中，

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

便可看到识别结果

3.5.3 HTTP服务调用

1.加载驱动,系统启动后加载一次即可（也可以加系统启动脚本）

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

若未加载驱动，可能报错：

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

设置系统时间（系统时间必须正确）

```
date --set "2020-7-28 20:48:00"
```

2.部署包中附带了HTTP服务功能，可直接运行

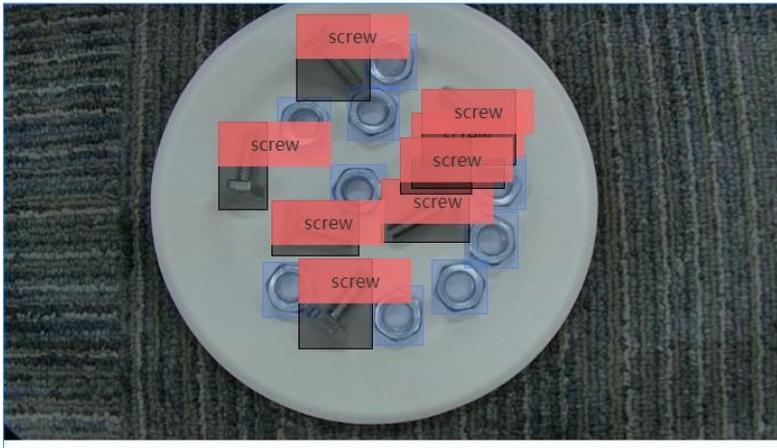
```
# ./easyedge_serving {RES目录} {序列号} {绑定的host, 默认0.0.0.0} {绑定的端口, 默认24401}
cd ${SDK_ROOT}
export LD_LIBRARY_PATH=./lib
./demo/easyedge_serving /home/root/workspace/EasyEdge-m1800-edgeboard/RES/ "1111-1111-1111-1111"
```

日志显示

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

则启动成功。此时可直接在浏览器中输入 `http://{FZ5 ip地址}:24401`，在h5中测试模型效果。

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 私有服务请求说明

- http 请求参数

URL中的get参数:

参数	说明	默认值
threshold	阈值过滤, 0~1	0.1

HTTP POST Body即为图片的二进制内容(无需base64, 无需json)

Python请求示例

```
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()
```

Java请求示例

- http 返回数据

字段	类型说明	其他
error_code	Number	0为成功,非0参考message获得具体错误信息
results	Array	内容为具体的识别结果。其中字段的具体含义请参考 预测图像-返回格式 一节
cost_ms	Number	预测耗时ms, 不含网络交互时间

返回示例

```
{
  "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
    }
  ]
}
```

- 错误说明

SDK所有主动报出的错误, 均覆盖在 `EdgeStatus` 枚举中。同时SDK会有详细的错误日志, 开发者可以打开Debug日志查看额外说明:

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

4.进阶指南

4.1 开发应用

4.1.1 模型获得

目前Paddle-Mobile仅支持Paddle训练的模型。如果你手中的模型是不同种类的模型，需要进行模型转换才可以运行。验证过的网络包含resnet、Inception、ssd、mobilenet等。

- 训练模型:
 1. 如果您没有模型，可以使用sample中的模型，或自己训练模型。详细使用参考[PaddlePaddle](#)
 2. 通过AI Studio平台训练模型，详细使用参考[AI Studio](#)
 3. 可以在EasyDL等平台上上传标注数据，训练模型，详细使用参考[EasyDL](#)
- 转换模型:
 1. 如果您已有caffe模型，我们提供了相应的转换工具，帮助转为Paddle模型。详细使用参考[X2Paddle_caffe2fluid](#)
 2. 如果您已有Tensorflow模型，我们提供了相应的转换工具，帮助转为Paddle模型。详细使用参考[X2Paddle_tensorflow2fluid](#)

4.1.2 连接视频数据源

FZ5提供多种视频输入硬件接口，支持多种协议输入图像数据作为数据源。包括HDMI、USB、GIGE等协议。

1. USB协议视频数据输入
可以选择uvc usb摄像头作为视频源。插入usb摄像头到FZ5 的usb接口
2. HDMI协议视频数据输入
可以选择HDMI接口作为视频源输入。将视频源输出端的HDMI OUT接口与FZ5的HDMI IN接口相连接。
3. GIGE协议视频数据输入
可以选择支持linux系统的GIGE相机，并联系我司进行相机官方sdk的适配，硬件连接FZ5的网口。

4.1.3 加载驱动

使用FZ5的加速功能，预测库会把计算量大的op通过驱动调用fpga进行运算。运行自己的应用前需要加载驱动，编译好的驱动位于/home/root/workspace/driver目录，提供无日志输出和有日志输出两个版本。

加载驱动

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

卸载驱动(正常情况您不需要卸载驱动，若需要加载有日志输出的版本，可以通过如下命令卸载后，再加载该版本)

```
rmmmod /home/root/workspace/driver/fpgadriv.ko
```

设置驱动自动加载

- 1) 在系统中添加自启动脚本

```
// 打开启动目录
cd /etc/init.d/
// 新建启动脚本并编辑, 名称可以自定义
vim eb.sh
```

脚本内容

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

2) 建立软链接

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

3) 更改脚本权限

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

4.1.4 使用预测库

FZ5支持Paddle-Moblie预测库, 编译好的预测库, 位于/home/root/workspace/paddle-mobile。具体使用 把预测库的头文件和动态库拷贝到自己应用中即可。另外可以参考我们提供的sample。Paddle-Moblie源码可以参考<https://github.com/PaddlePaddle/paddle-mobile>

4.1.5 创建应用

4.1.5.1 添加预测库

拷贝/home/root/workspace/paddle-mobilie/下面的动态库和头文件到您的工程中。在CmakeLists.txt添加paddle-mobile库的引用

```
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 添加模型

拷贝自己训练的模型到您的工程中

4.1.5.3.添加预测数据源

可以选择图片, 摄像头数据或者HDMI视频数据作为预测数据源, 使用摄像头需要插入USB摄像头, 使用HDMI作为数据源需要将视频输出端的HDMI OUT接口与FZ5的HDMI IN接口相连接。

- USB摄像头

1) 插入摄像头后, 通过ls /dev/video* 查看设备接入情况。通过会显示如下:

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

/dev/video2为usb摄像头v4l2输出yuv数据, 当应用提示找不到设备时, 可以修改src/video_classify.cpp或者src/video_detection.cpp。通过/home/root/workspace/tools下video工具检测摄像头联通性

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

2) 另外可以修改摄像头分辨率

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

3) 运行video工具

```
//读取usb摄像头, 采集一张图片保存到本地  
cd /home/root/workspace/tools/video  
./v4l2demo -i /dev/video2 -j -n 1  
//如有疑问, 查看帮助  
./v4l2demo -h
```

执行程序后在build目录下会生成jpg文件, 可以查看图片否正确。如果没有生成图片, 检测是否识别到USB设备。

- HDMI视频源

插入HDMI视频源后, 通过/home/root/workspace/tools下video工具检测HDMI接口的联通性

1) 查看设备, 正常情况下设备为/dev/vide01

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

2) 配置视频源分辨率参数

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

3) 运行video工具

```
//读取HDMI视频源数据, 采集一张图片保存到本地  
cd /home/root/workspace/tools/video  
./v4l2demo -i /dev/video1 -j -n 1  
//如有疑问, 查看帮助  
./v4l2demo -h
```

执行程序后在build目录下会生成jpg文件，可以查看图片否正确。如果没有生成图片，请检查HDMI连接是否正确。

4.1.5.4 调用预测库加载模型和使用预测数据

- 初始化模型

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

- 准备数据
 - 1.缩放图片到指定的大小。如果网络只能固定大小输入，需要缩放到网络输入大小。
 - 2.图片预处理（减均值、转浮点、归一化等）。
 - 3.产出数据，由于FZ5使用的NHWC格式，通常视频过来的数据就是NHWC格式，就不需要NHWC->NCHW转换。
- 预测数据

调用API的predict接口，传输处理好的数据，获取预测结果

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

4.2 驱动说明

驱动模块包含设备管理、内存管理、IO设定、参数管理及命令控制等功能

1.设备管理

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

2.内存管理

驱动部分会为自己及FPGA设备从系统内存中保留自己专用的内存，这部分内存操作系统Linux在没有Driver的情况下看不到或不可独立操作，需要借助驱动的功能来使用，来完成相应内存管理（分配、释放、映射、拷贝等功能）

```
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设定、参数管理及命令控制

同时驱动还为上层（顶层的主控应用）提供IO设定、参数管理、及命令控制等功能，并完成相应的、和FPGA设备的通信设定。

```
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 预测库说明

1.Paddle-Mobile

Paddle-Mobile是PaddlePaddle组织下的项目，是一个致力于嵌入式平台的深度学习预测框架。FZ5使用的是[Paddle-Mobile](#)下的FPGA实现的预测库。

2.NHWC

基于FPGA的特性，Paddle-Mobile的FPGA实现数据格式为NHWC。在开发自己的应用或修改paddle-mobile代码时，需要注意。

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

3.FP16(实现为half)

FPGA实现的OP, Tensor数据为FP16。当遇到CPU实现的OP时，需要把FP16转换成FP32(即Float)。反之，需要把FP32转换成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.对齐

基于FPGA的特性，使用FPGA实现的OP前需要将Tensor的数据基于C*W进行16位倍数进行对齐。反之，上一个节点是FPGA实现，下一个节点为CPU实现时需要反对齐

```
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定制

当Paddle-Mobile现有的OP,无法满足您的模型需求时，可以新增或定制OP。具体可以参考[Paddle-Mobile OP代码设计](#)。OP定制需要主要NHWC、FP16、对齐等问题。

附录一售后服务与技术支持

凡是通过米尔科技直接购买或经米尔科技授权的正规代理商处购买的米尔科技全系列产品，均可享受以下权益：

- 1、6个月免费保修服务周期
- 2、终身免费技术支持服务
- 3、终身维修服务
- 4、免费享有所购买产品配套的软件升级服务
- 5、免费享有所购买产品配套的软件源代码，以及米尔科技开发的部分软件源代码
- 6、可直接从米尔科技购买主要芯片样品，简单、方便、快速；免去从代理商处购买时，漫长的等待周期
- 7、自购买之日起，即成为米尔科技永久客户，享有再次购买米尔科技任何一款软硬件产品的优惠政策
- 8、OEM/ODM 服务

如有以下情况之一，则不享有免费保修服务：

- 1、超过免费保修服务周期
- 2、无产品序列号或无产品有效购买单据
- 3、进液、受潮、发霉或腐蚀
- 4、受撞击、挤压、摔落、刮伤等非产品本身质量问题引起的故障和损坏
- 5、擅自改造硬件、错误上电、错误操作造成的故障和损坏
- 6、由不可抗拒自然因素引起的故障和损坏

产品返修：用户在使用过程中由于产品故障、损坏或其他异常现象，在寄回维修之前，请先致电米尔科技客服部，与工程师进行沟通以确认问题，避免故障判断错误造成不必要的运费损失及周期的耽误。

维修周期：收到返修产品后，我们将即日安排工程师进行检测，我们将在最短的时间内维修或更换并寄回。一般的故障维修周期为3个工作日（自我司收到物品之日起，不计运输过程时间），由于特殊故障导致无法短期内维修的产品，我们会与用户另行沟通并确认维修周期。

维修费用：在免费保修期内的产品，由于产品质量问题引起的故障，不收任何维修费用；不属于免费保修范围内的故障或损坏，在检测确认问题后，我们将与客户沟通并确认维修费用，我们仅收取元器件材料费，不收取维修服务费；超过保修期限的产品，根据实际损坏的程度来确定收取的元器件材料费和维修服务费。

运输费用：产品正常保修时，用户寄回的运费由用户承担，维修后寄回给用户的费用由我司承担。非正常保修产品来回运费均由用户承担。

购买请联系：

电话：0755-25622735

传真：0755-25532724

邮箱：sales@myirtech.com

网站：www.myir-tech.com

技术支持请联系：

电话：0755-25622735

传真：0755-25532724

邮箱：support@myirtech.com

网站：www.myir-tech.com