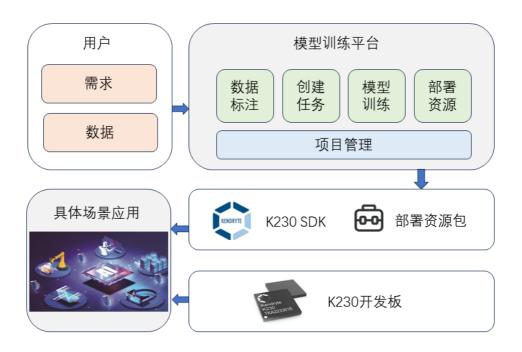


在线训练平台使用文档

V 1.1

1.平台介绍

嘉楠科技开发者社区(https://developer.canaan-creative.com/) 模型训练板块是为简化开发流程,提高开发效率开放的训练平台。该平台使用户更加注重视觉场景的落地实现,更加快捷的完成从数据标注到获得部署包,并在搭载嘉楠科技Kendryte®系列AIoT芯片中最新一代SOC产品K230的开发板上进行部署的全部过程。该平台现支持图像分类和图像检测两类任务。



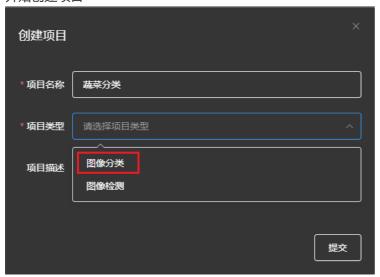
模型训练平台对接用户和板上开发人员,通过创建项目对具体场景下的AI落地过程进行管理,导入数据后需要用户创建任务并配置训练参数,然后一键启动训练,训练结束后得到部署资源包。部署资源包可以和K230 SDK结合使用,完成后续的开发板开发工作,实现模型部署。

2.创建项目

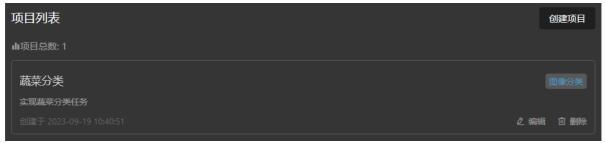
进入嘉楠科技开发者社区模型训练板块: https://developer.canaan-creative.com/model/training



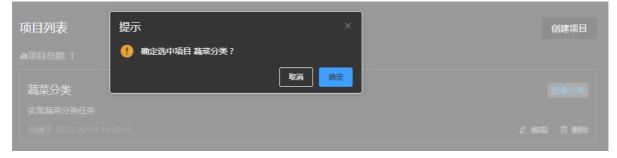
开始创建项目



项目创建完成后在项目列表可以看到新建的项目。



点击项目列表中的新创建项目条目,确定当前选中项目。



3.上传数据集

3.1 图像分类

创建数据集





绑定新建数据集到当前项目

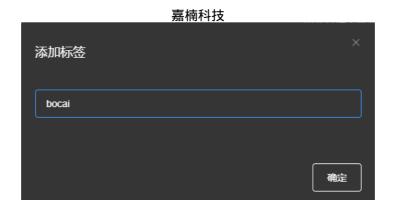


采集数据





version 1.1





分类任务要求至少上传两个类别。

3.2 图像检测

图像检测任务和图像分类任务的项目创建方式一致。创建一个昆虫检测项目,绑定数据集。



数据上传方式包括两种方式:

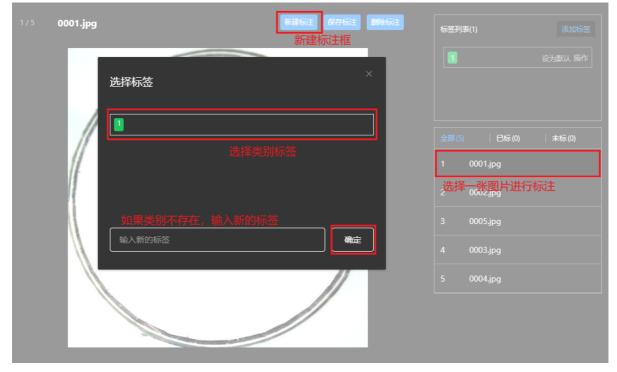


3.2.1 图片上传



添加标签

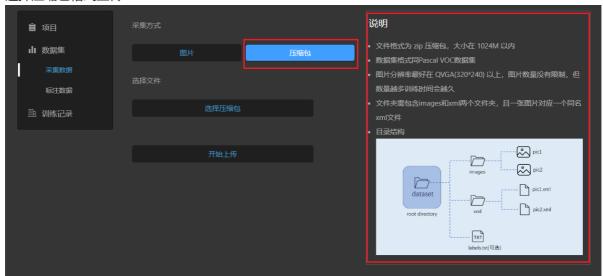




完成当前图片标注后保存标签。标注完所有图片后启动训练。

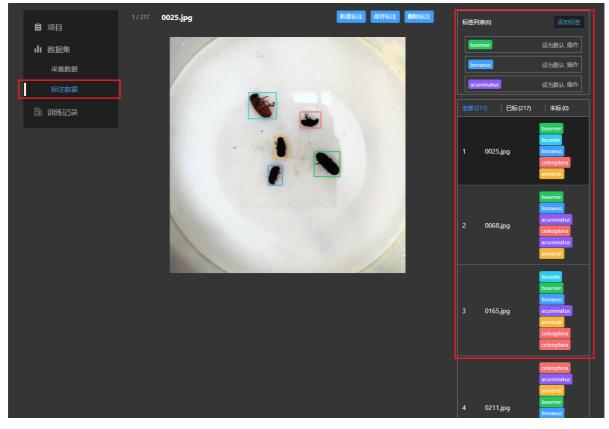
3.2.2 zip压缩包上传

选择压缩包格式上传



压缩包必须按照图示格式进行组织。

查看导入的图片标注信息。



4.新建任务



设置训练参数





点击详情进入训练界面



训练过程中输出日志和损失曲线等信息。

5.下载部署包

训练结束后, 在训练任务条目中选择下载按钮



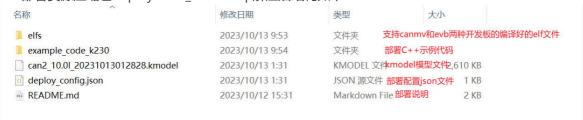
下载得到部署资源包。

部署资源包内包含:测试结果图片、生成的kmodel、部署配置文件deploy_config.json、部署资源压缩 包。

部署资源压缩包解压后,参照README.md可以实现上板部署过程。



C++部署资源压缩包deployment_source.zip解压后结构如下:



嘉楠科技

MicroPython部署资源压缩包mp_deployment_source.zip解压后结构如下:

| 名称 个 | 修改日期 | 类型 | 大小 | |
|---------------------------------|------------------|---------------|----------|-----------|
| an2_10.0l_20231123111851.kmodel | 2023/11/23 11:20 | KMODEL 文件 | 2,610 KB | kmodel |
| 🖺 cls_image.py | 2023/11/7 16:57 | PY 文件 | 4 KB | 静态图推理脚本 |
| 🖺 cls_video.py | 2023/11/7 16:59 | PY 文件 | 7 KB | 视频流推理脚本 |
| deploy_config.json | 2023/11/23 11:20 | JSON 源文件 | 1 KB | 部署配置文件 |
| README.md | 2023/11/7 11:00 | Markdown File | 2 KB | readme.md |

6.上板部署--C++

若使用MicroPython的部署包请参见下一章节。

注意:训练环境中nncase和nncase-kpu的版本和SDK的版本要对应,nncase和nncase-kpu版本为2.4.0,SDK版本为1.1。

上板部署过程分为以下几个步骤:

6.1 板上系统镜像编译

K230 SDK需要在*Linux环境*下编译,推荐使用Ubuntu Liunx 20.04。 使用docker编译环境,下载 k230 sdk。

1 # 下载docker编译镜像 docker pull ghcr.io/kendryte/k230_sdk # 可以使用以下命令确认docker镜像拉取成功 3 docker images | grep k230_sdk # 下载sdk 5 git clone -b v1.1 --single-branch https://github.com/kendryte/k230_sdk.git 7 cd k230_sdk # 下载工具链, make prepare_sourcecode 会自动下载Linux和RT-Smart toolchain, buildroot package, AI package等. 请确保该命令执行成功并没有Error产生,下载时间和速度 以实际网速为准。 make prepare_sourcecode 9 10 # 创建docker容器, \$(pwd):\$(pwd)表示系统当前目录映射到docker容器内部的相同目录下,将系 统下的工具链目录映射到docker容器内部的/opt/toolchain目录下 docker run -u root -it -v \$(pwd):\$(pwd) -v \$(pwd)/toolchain:/opt/toolchain -11 w \$(pwd) ghcr.io/kendryte/k230_sdk /bin/bash

K230现有两种开发板,分别为CANMV-K230-V1.0(以下简称CANMV-K230)和K230-USIP-LP3-EVB (以下简称K230-EVB)。两种开发板区别如图:

K230-USIP-LP3-EVB







- # 在docker中编译镜像,请耐心等待完成,不同类型开发板编译命令不同
 # 如果是CANMV-K230开发板
 make CONF=k230_canmv_defconfig
 # 如果是K230-EVB开发板
 make CONF=k230_evb_defconfig
- SD卡镜像也可在嘉楠开发者社区资料下载板块找到。

6.2 系统镜像烧录

CANMV-K230开发板:

编译结束后在output/k230_canmv_defconfig/images目录下可以找到编译好的镜像文件:

```
1 k230_canmv_defconfig/images
2 ├─ big-core
3 ├─ little-core
4 ├─ sysimage-sdcard.img # SD卡镜像
5 └─ sysimage-sdcard.img.gz # SD卡镜像压缩包
```

CANMV-K230开发板支持SD卡镜像启动。

K230-EVB开发板:

编译结束后在output/k230_evb_defconfig/images目录下可以找到编译好的镜像文件:

```
1 k230_evb_defconfig/images
2 ├─ big-core
3 ├─ little-core
4 ├─ sysimage-sdcard.img # 镜像
5 └─ sysimage-sdcard.img.gz # 镜像压缩包
```

K230 支持SDCard、eMMC、norflash等多种启动方式。

烧录TF卡

详细烧录步骤参考K230 SDK 使用说明。

Linux: 如使用Linux烧录TF卡,需要先确认SD卡在系统中的名称/dev/sdx,并替换如下命令中的/dev/sdx

1 sudo dd if=sysimage-sdcard.img of=/dev/sdx bs=1M oflag=sync

Windows: 如使用Windows烧录, 建议使用<u>the balena Etcher</u>工具。将生成的sysimage-sdcard.img下载到本地,使用烧录工具<u>the balena Etcher</u>进行烧录。



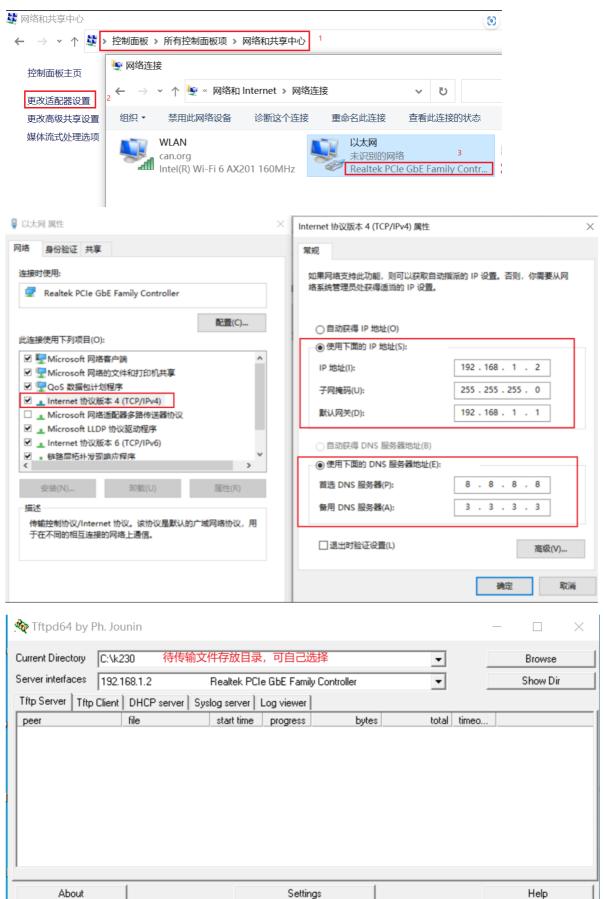
其它更详细的烧录方法,请参考K230 SDK 使用说明。

确认启动开关选择在SD卡启动模式下,将烧录完成的TF卡插入进开板板卡槽中,然后将电源开关K1拔到ON位置,系统可上电。如果您有接好串口,可在串口中看到启动日志输出。 **系统上电后,默认会有二个串口设备,可分别访问小核Linux和大核RTSmart** 大核RTSmart系统中会开机会自动启动一个应用程序,可按q键退出至命令提示符终端。

6.3 搭建文件传输环境

windows系统

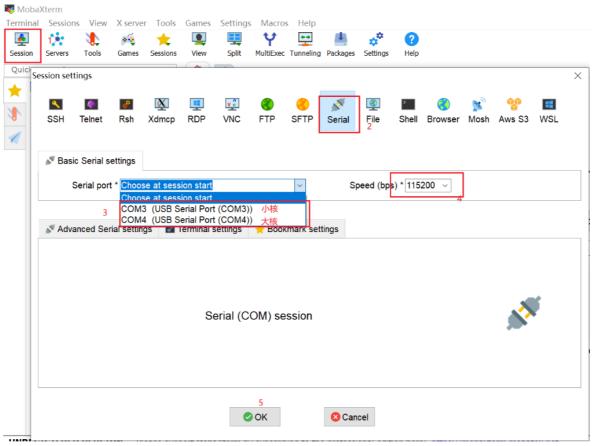
- (1) Tftpd64安装,在https://bitbucket.org/phjounin/tftpd64/downloads/下载。
- (2) MobaXterm安装:在https://mobaxterm.mobatek.net/download.html下载安装。
- (2) 配置PC网络:



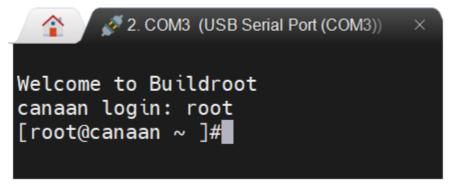
(4) 开发板网络配置:

开发板上电,电源线、网线、COM口连接线配置见文档: <u>K230 SDK 使用说明</u>。打开MobaXterm,通过两路COM串口连接开发板,COM编号不固定,较小为小核串口,较大为大核串口。





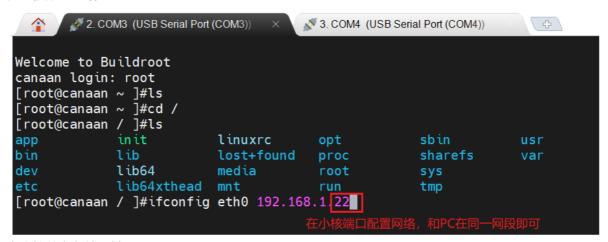
小核进入后回车,进入如下界面,使用root登录:



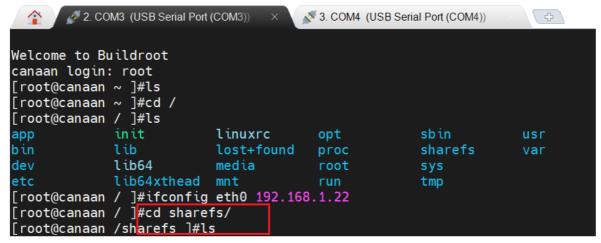
大核进入后回车,进入如下界面:



在小核配置网络:



大小核共享存储区域: /sharefs



当要从Tftpd64配置的文件中拷贝数据时,在小核界面使用如下命令:

#192.168.1.2 为PC的局域网IP

```
1 # 192.168.1.2 为PC的局域网IP
2 tftp -g -r your_file_name 192.168.1.2
```

当将开发板文件拷贝到PC端Tftpd64配置的文件夹下时,在小核使用如下命令:

```
1 # 192.168.1.2 为PC的局域网IP
2 tftp -p -r your_file_name 192.168.1.2
```

Linux系统

在Linux系统中,PC正常连接网络,开发板可以通过网线连接PC所在网关下其他网口,通过scp命令实现文件传输。

开发板上电,进入大小核COM界面,在小核执行scp传输命令:

```
1 # 从PC拷贝文件至开发板
2 scp 用户名@域名或IP:文件所在目录 开发板目的目录
3 # 从开发板拷贝文件至PC
4 scp 开发板待拷贝目录 用户名@域名或IP:PC目的目录
```

6.4 部署代码编译elf文件

将部署资源压缩包deployment_source.zip解压,阅读README.md,将子文件夹example_code_k230 拷贝到SDK文件目录下的src/big/nncase子目录下,授予权限,编译elf可执行文件。

编译K230-EVB开发板的可执行文件:

```
1 | chmod +x build_app.sh
2 | ./build_app.sh evb
```

编译结束后在example_code_k230文件夹下的k230_bin中可以获得编译完成的main.elf文件。

编译CANMV-K230开发板的可执行文件:

```
chmod +x build_app.sh
logical chmod +x build_app.sh
logical chmod +x build_app.sh
```

6.5 文件传输

在小核/sharefs下新建一子目录test_deploy:

```
1 cd /sharefs
2 mkdir test_deploy
3 cd test_deploy
```

通过tftp将部署资源包中的kmodel文件、deploy_config.json文件、main.elf文件和待测试图片文件拷贝到小核/sharefs/test_deploy目录下。

6.6 上板调试

上板调试命令:

```
# 进入大核对应目录下,执行main.elf deploy_config.json 图片/None debug_mode deploy_config.json是部署配置文件 # 图片/None 如果是图片路径,则执行静态图推理,如果调用摄像头进行视频流推理,此参数为None debug_mode: 调试模式,有0、1、2三个选项; 分别对应不调试、简单调试、详细调试
```

在大核中进入/sharefs/test_deploy目录下,执行main.elf可执行文件。

```
1
# 静态图推理调试

2
main.elf deploy_config.json test.jpg 1

3
# 视频流推理调试

4
main.elf deploy_config.json None 1
```

7.上板部署--MicroPython

注意:训练环境中nncase和nncase-kpu的版本和SDK的版本要对应,nncase和nncase-kpu版本为2.4.0,SDK版本为1.1。

上板部署过程分为以下几个步骤:

7.1板上系统镜像编译

CanMV 的目的是让 AIOT 编程更简单,基于 <u>Micropython</u> 语法, 运行在<u>Canaan</u>强大的嵌入式AI SOC系列上。目前它在K230上运行。使用docker编译环境,下载<u>k230 CanMV</u>。

```
git clone https://github.com/kendryte/k230_canmv.git
cd k230_canmv
make prepare_sourcecode
# 生成docker镜像(第一次编译需要,已经生成docker镜像后跳过此步骤,可选)
docker build -f k230_sdk/tools/docker/Dockerfile -t k230_docker
k230_sdk/tools/docker
# 启动docker环境(可选)
docker run -u root -it -v $(pwd):$(pwd) -v
$(pwd)/k230_sdk/toolchain:/opt/toolchain -w $(pwd) k230_docker /bin/bash
# 默认使用canmv板卡,如果需要使用其他板卡,请使用 make CONF=k230_xx_defconfig,支持的
板卡在configs目录下
make
```

7.2 镜像烧录

参见6.2章节。

7.3搭建文件传输环境

参见6.3章节。

7.4文件拷贝

离线拷贝文件,使用读卡器将kmodel文件和deploy_config.json文件拷贝到sdcard目录下,若进行静态图推理也可拷贝一张测试图片。记住文件所在目录。

7.5IDE安装

您可以下载CanMV IDE,请选择4.0.5版本进行安装以支持K230:

官网下载: https://kendryte-download.canaan-creative.com/developer/common/canmv-ide-4.0.5.e xe.ow/developer/common/canmv-ide-4.0.5.e

github下载: Releases · kendryte/canmv ide (github.com)。

7.6上板调试

按下图步骤,建立连接。点击如下按钮:



选择除大小核串口外的第三个串口:



连接建立后,选择文件->打开文件,选择解压的部署包mp_deployment_source中的##_image.py和##__video.py,编辑更改kmodel和deploy_config.json文件的拷贝路径。

```
root_path="/sdcard/app/tests/test_cls/"
config_path="sdcard/app/tests/test_cls/deploy_config.json"
image_path="/sdcard/app/tests/test_cls/test_cls.jpg"
deploy_conf-{}
deploy_conf-{}
debug_mode=1
```

点击左下角的绿色三角按钮运行,点击串行终端查看命令行输出。

工具

烧录工具the balena Etcher: https://etcher.balena.io/

局域网文件传输工具Tftpd64: https://bitbucket.org/phjounin/tftpd64/downloads/

MobaXterm下载地址: https://mobaxterm.mobatek.net/download.html

CANMV-K230开发板1.1版本SD卡镜像下载: https://kendryte-download.canaan-creative.com/developer/k230/k230 canmv sdcard V1.1 nncase V2.4.0.img,gz

K230-EVB开发板1.1版本SD卡镜像下载: https://kendryte-download.canaan-creative.com/develope
r/k230/k230 evb sdcard V1.1 nncase V2.4.0.img.gz

CANMV-K230 MicroPython1.1版本SD卡镜像下载: https://kendryte-download.canaan-creative.co m/developer/k230/CanMV-K230 micropython v0.3 sdk v1.1 nncase v2.4.0.img,gz

CanMV IDE下载: Releases · kendryte/canmv ide (github.com)

参考

k230_sdk github: https://github.com/kendryte/k230 sdk

k230_sdk_doc github: k230 sdk 使用说明

k230_sdk gitee: https://gitee.com/kendryte/k230_sdk

k230_canmv github:kendryte/k230_canmv (github.com)

nncase github: <u>kendryte/nncase</u>: <u>Open deep learning compiler stack for Kendryte AI accelerator</u>

(github.com)