

关于本手册

本文介绍了如何上手使用PaddlePi-K210开发套件。

发布说明

日期	版本	发布说明
2020.1.20	V0.1	首次发布
2020.5.7	V0.2	更新示例链接，修正部分描述

免责声明

本文中的信息，包括参考的 URL 地址，如有变更，恕不另行通知。文档“按现状”提供，不负任何担保责任，包括对适销性、适用于特定用途或非侵权性的任何担保和任何提案、规格或样品在他处提到的任何担保。

本文档不负任何责任，包括使用本文档内信息产生的侵犯任何专利权行为的责任。本文档在此未以禁止反言或其他方式授予任何知识产权使用许可，不管是明示许可还是暗示许可。

文中提到的所有商标名称、商标和注册商标均属其各自所有者的财产，特此声明。

版权公告

版权归 © 2020 嘉楠科技所有。保留所有权利。

1.产品介绍

PaddlePi-K210是百度和嘉楠联合推出的一款深度开发套件，该套件包含了AI Studio模型训练、应用开发、模型部署等整套解决方案。可用于深度学习相关的场景教学，以及工业、农业深度学习相关场景的应用，如人脸检测、病虫害监控和工业质检。本教程将介绍如何上手并使用PaddlePi-K210开发套件进行应用开发。

1.1 硬件介绍

PaddlePi-K210是基于嘉楠科技AI芯片-勘智K210打造的一款深度学习开发套件，该芯片具有如下特点：

1、双核RISC-V 64位处理器

- 高达400MHZ的主频：轻松应对各种业务场景及计算任务。
- 硬件浮点指令加速：两个核心均支持单精度指令加速，可使简单的渲染高达105fps。
- 双核64位处理器：一个计算核，一个应用核，使开发者开发复杂应用场景时资源不再捉襟见肘。
- 芯片功耗小于350mW，典型应用场景功耗小于1w。

2、端侧常见图像任务的神经网络推理

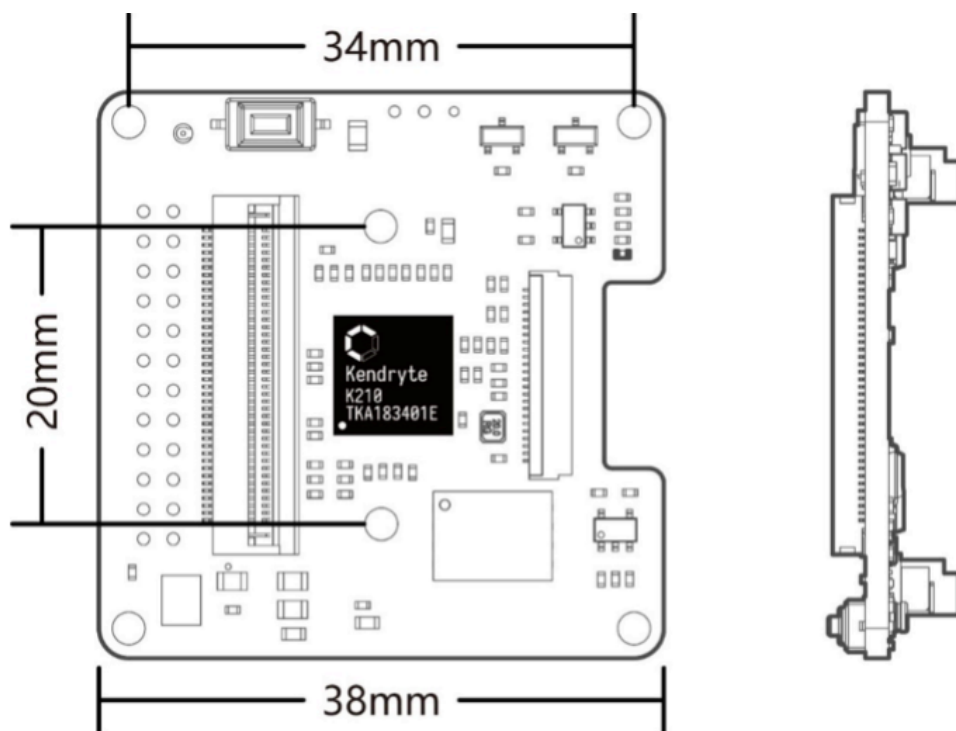
- 图像分类：可达到最快120fps的分类卷积神经网络。
- 目标检测：QVGA的网络输入下，目标检测模型帧率可达到60fps。
- 传统算法加速：不仅可以加速卷积神经网络的推理，并且可以对传统图像算法基于卷积的算子进行有效的加速。

3、硬件组成

PaddlePi-K210主板、DVP 200W摄像头、LCD触摸显示屏和调试器板。

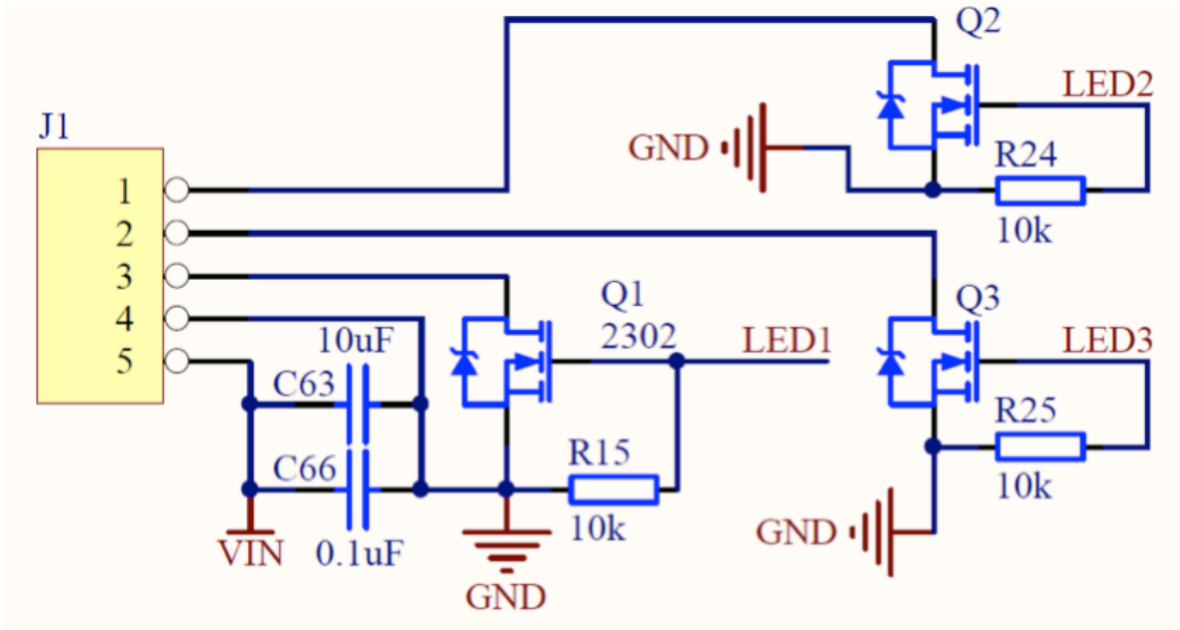
开发板板载摄像头接口，LCD触摸屏接口，拓展接口，WiFi和SPI Flash。

1) 主板结构尺寸



2) 接口功能介绍

外部LED接口

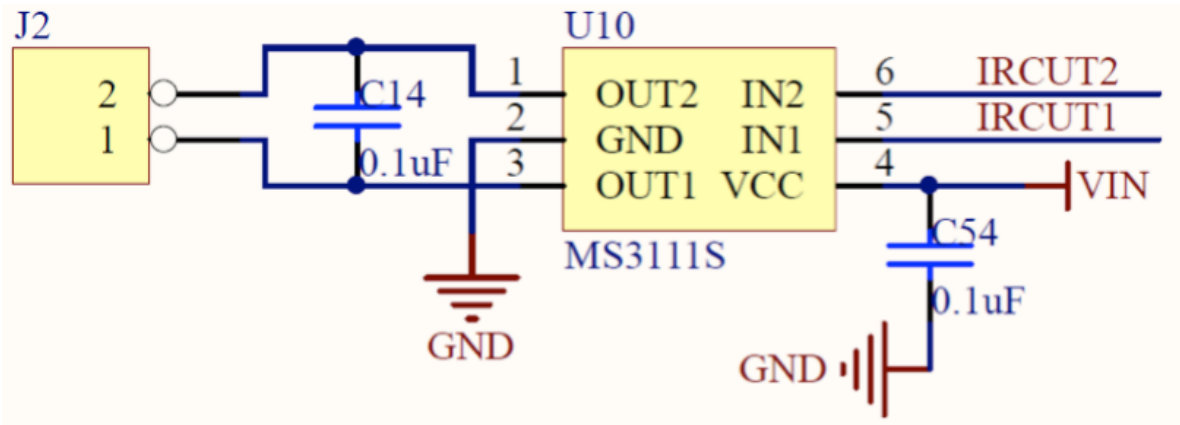


J1接口各管脚功能如下表：

标号	功能	说明	备注
1	LED2	接外部指示灯	电流小于100mA
2	LED3	接外部指示灯	电流小于100mA
3	LED1	接外部红外补光灯	电流<=1A
4	GND	地	
5	5V	电源输出	

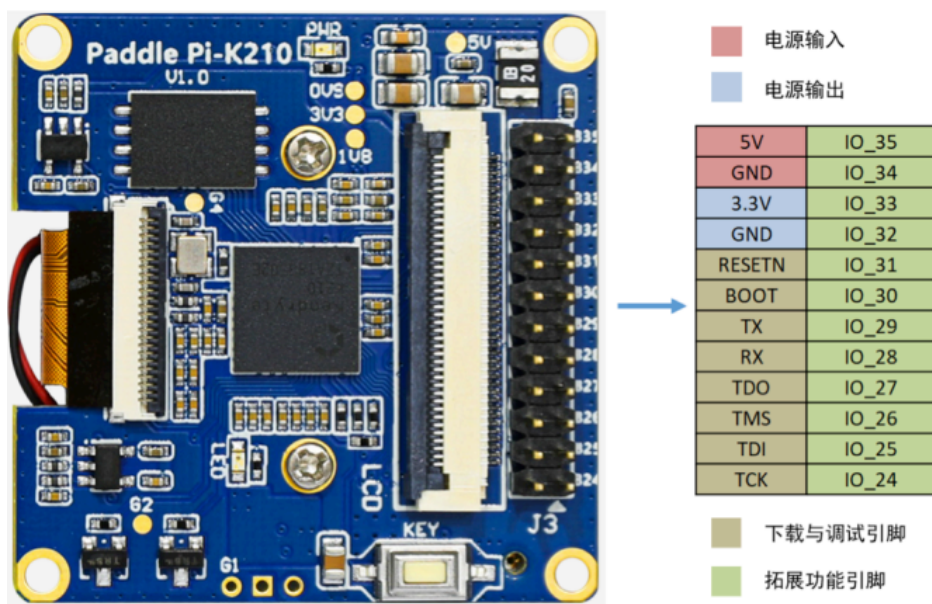
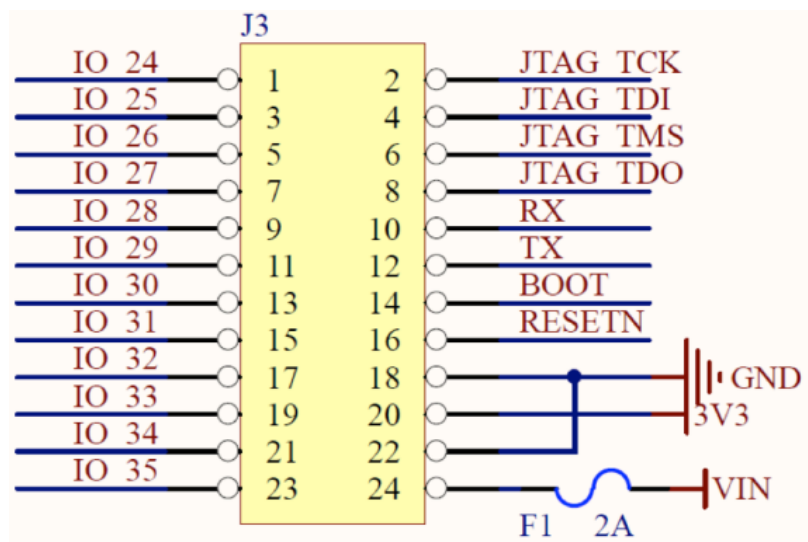
其中4号和5号管脚对外输出5V电源。3号管脚设计为外接红外补光灯用，也可做其他驱动用，注意电流不要超过1A。1号和2号管脚设计为驱动两路独立的指示灯用，也可做其他驱动使用，注意电流不要超过100mA。

IR-CUT接口



J2接口设计为与IR-CUT摄像头对接，驱动IR-CUT使用。不推荐做其他功能使用。

拓展接口（所有IO管脚电平为3.3V）



J3为拓展接口，其各引脚功能如图所示。在电源输入管脚处使用5V输入，板子即可上电正常工作。

标号	类型	说明	备注
1	I/O	输入输出引脚24	
2	I/O	JTAG_TCK	
3	I/O	输入输出引脚25	
4	I/O	JTAG_TDI	
5	I/O	输入输出引脚26	
6	I/O	JTAG_TMS	
7	I/O	输入输出引脚27	
8	I/O	JTAG_TDO	
9	I/O	输入输出引脚28	
10	I/O	UARTHS_RX	
11	I/O	输入输出引脚29	
12	I/O	UARTHS_TX	
13	I/O	输入输出引脚30	
14	I/O	BOOT	
15	I/O	输入输出引脚31	
16	---	RESET	
17	I/O	输入输出引脚32	
18	POWER	GND	
19	I/O	输入输出引脚33	
20	POWER	3.3V	电源输出
21	I/O	输入输出引脚34	
22	POWER	GND	
23	I/O	输入输出引脚35	
24	POWER	5V	电源输入

注：单独使用22和24两个管脚供电时，注意检查电源顺序，禁止反接，否则会烧毁供电电路。

1.2 软件介绍

百度大脑AI Studio(训练完成的模型在PaddlePi-K210上运行)

AI Studio是针对AI开发者的在线一体化开发平台。该平台集成了AI教程、深度学习样例工程、各领域经典数据集、强大的云端算力资源及存储资源，从而解决了开发者在AI学习过程中的一系列难题，例如样例代码难以直接应用、高质量的数据集不易获得以及本地环境难以使用大体量数据集进行模型训练等等。

百度AI Studio平台已经为使用者预置了Python语言环境，以及百度PaddlePaddle深度学习开发框架。

同时该平台还提供了PaddlePi-K210开发套件官方项目，开发者只需要复制相应的项目，重新训练，即可获取模型。您也可以修改成自己的数据集，进行训练，以适配您的应用场景。

AI Studio基本使用可以参考：[AI Studio](#)

PaddlePi-K210工具链

我们提供了PaddlePi-K210开发所需要的 SDK、交叉编译工具、调试工具、IDE，并且提供了 AI 加速器部分所需要的模型转换器(Model Compiler)。详细可以访问<https://kendryte.com/downloads/>和<https://github.com/kendryte>。

操作系统：裸机或FreeRTOS

支持网络模型： Mobilenet、TinyYOLOv2 (after pruned)

2. 基本使用

2.1 训练模型

AI Studio平台针对PaddlePi-K210提供了mobilenet 和 tiny yolo网络的官方示例，分别是[PaddlePi-mobilenet](#)和[PaddlePi-tiny-yolo](#)，只需要fork过来就行，无需自行搭建网络。

2.1.1 fork AI Studio平台官方项目PaddlePi-mobilenet训练模型

以下以"PaddlePi-mobilenet"为例进行讲解，该示例训练一个花的5分类网络，可以根据自己的需求替换数据集。

1. 点击[PaddlePi-mobilenet](#)进入。
2. 运行项目，可以选择CPU或GPU模型。
3. 开始训练模型，按顺序运行cell即可。

- 1) 运行cell，解压数据集

```
# 解压花朵数据集
!cd data/data2815 && unzip -q flower_photos.zip
```

- 2) 运行cell，解压预训练模型参数

```
# 解压预训练模型参数
!cd data/data6592 && unzip -q MobileNetV2_pretrained.zip
```

3) 运行cell，预处理数据，将其转化为标准格式。同时将数据拆分成两份，以便训练和计算预估准确率。

```
# 预处理数据，将其转化为标准格式。同时将数据拆分成两份，以便训练和计算预估准确率
import codecs
import os
import random
import shutil
from PIL import Image

train_ratio = 4 / 5

...
省略
...

train_file.close()
eval_file.close()
```

- 4) 运行cell，构建网络并训练模型

```
# -*- coding: UTF-8 -*-
"""
训练常用视觉基础网络，用于分类任务
需要将训练图片，类别文件 label_list.txt 放置在同一个文件夹下
```

程序会先读取 `train.txt` 文件获取类别数和图片数量

```
"""
from __future__ import absolute_import
from __future__ import division
from __future__ import print_function
import os
import numpy as np
import time
import math
import paddle
import paddle.fluid as fluid
import codecs
import logging

from paddle.fluid.initializer import MSRA
from paddle.fluid.initializer import Uniform
from paddle.fluid.param_attr import ParamAttr
from PIL import Image
from PIL import ImageEnhance

...
省略
...

if __name__ == '__main__':
    init_log_config()
    init_train_parameters()
    train()
```

日志提示如下，表示训练完成。选择CPU训练，可能需要几个小时。

```
[line:635] - INFO: end training
[line:649] - INFO: training till last epcho, end training
```

5) 运行cell，评估模型

```
from __future__ import absolute_import
from __future__ import division
from __future__ import print_function

import os
import numpy as np
import random
import time
import codecs
import sys
import functools
import math
import paddle
import paddle.fluid as fluid
from paddle.fluid import core
from paddle.fluid.param_attr import ParamAttr
from PIL import Image, ImageEnhance

...
省略
```



```
...

if __name__ == '__main__':
    eval_all()
```

使用测试数据评估效果，运行输出：

```
total eval count:750 cost time:65.95 sec predict accuracy:0.84
```

6) 模型转换（普通的模型并不能很好地运行在开发板等特定硬件上，为了在特定硬件上部署，需要借助一些工具）。首先拉取并解压模型转换工具：

```
!mkdir /home/aistudio/work/ncc
!wget "https://platform.bj.bcebos.com/sdk%2Fncc-linux-x86_64.tar.gz" -O ncc-
linux-x86_64.tar.gz
!tar -zxvf ncc-linux-x86_64.tar.gz -C /home/aistudio/work/ncc
```

7) 进行模型压缩，我们需要进行量化。为了保证量化后的精度，需要使用训练图片调整模型。拷贝评估图片到/home/aistudio/work/images

```
!mkdir /home/aistudio/work/images

import os
import shutil
filenames = os.listdir("/home/aistudio/data/data2815/evalImageSet/")

index = 0
for i in range(1, len(filenames), 7):
    srcFile = os.path.join("/home/aistudio/data/data2815/evalImageSet/",
filenames[index])
    targetFile = os.path.join("/home/aistudio/work/images", filenames[index])
    shutil.copyfile(srcFile, targetFile)
    index += 7
```

8) 进行模型转换，把训练好的模型转换成PaddlePi-K210可以运行的模型。

```
!/home/aistudio/work/ncc/ncc -i paddle -o k210model -p /dev/ttyUSB0 --
postprocess n1to1 --dataset work/images/ freeze-model mobilenet.kmodel
```

日志显示如下，表示已经运行完成

```
KPU memory usage: 2097152 B
Main memory usage: 250880 B
```

[nncase](#) is a cross-platform neural network optimization toolkit for fast inference.

```
ncc -i <input format> -o <output format> [--dataset <dataset path>] [--postprocess
<dataset postprocess>] [--weights-bits <weights quantization bits>] <input path>
<output path>
```

Parameter	value	description
-i	paddle	input format <code>__model__</code> Paddle model
-o	k210model	ouput format <code>.kmodel</code> K210 model (Only supported in inference mode)
--dataset	eg:work/images/	eval images path, required when the output format is <code>k210model</code> or <code>inference</code> .
--postproces	0to1	normalize images to [0, 1]
--postproces	n1to1	normalize images to [-1, 1]
--weights-bits	8	8bit quantization [0, 255]
--weights-bits	16	16bit quantization [0, 65535]
--float-fc		Use float fullyconnected kernels.
--channelwise-output		Use channelwise quantization for output layers.

Supported layers

layer	parameters
Conv2d	kernel={3x3,1x1} stride={1,2} padding=same *
DepthwiseConv2d	kernel={3x3,1x1} stride={1,2} padding=same *
FullyConnected	
Add	
MaxPool2d	
AveragePool2d	
GlobalAveragePool2d	
BatchNormalization	
BiasAdd	
Relu	
Relu6	
LeakyRelu	
Concatenation	
L2Normalization	

layer	parameters
Sigmoid	
Softmax	
Flatten	
ResizeNearestNeighbor	

2.1.2 fork AI Studio平台官方项目PaddlePi-tiny-yolo训练模型

以下以“PaddlePi-tiny-yolo”为例进行讲解，该示例是训练一个螺丝螺母目标检测的网络，可以根据自己的需求替换数据集。

1. 点击[PaddlePi-tiny-yolo](#)进入。
2. 运行项目，可以选择CPU或GPU模型。
3. 开始训练模型，按顺序运行cell即可。

- 1) 运行cell，解压数据集。
- 2) 将数据集处理成需要的格式：

```
* label_list.txt 每一行一个类别，类别编号 类别名字
* train.txt 每一行一个样本，图片路径[{"bbox":{"left":0, "top":0, "width": 0,
"height": 0}, "label":1}...]
* trainImageSet/xxx.jpg 训练图片
* eval.txt 格式同train.txt
* evalImageSet/xxx.jpg 验证图片
```

- 3) 配置tiny yolo模型的参数。
- 4) 搭建tiny yolo网络。
- 5) 训练网络。
- 6) 模型转换。

2.2 编译和烧录程序

将AI Studio上训练的模型下载，拷贝到示例代码中进行编译并烧录程序。（关于PaddlePi-K210开发环境的搭建，请参考文档《PaddlePi-K210命令行开发环境搭建指南》）。

2.2.1 PaddlePi-mobilenet编译和烧录程序

windows环境

交叉编译环境和SDK 参考文档《PaddlePi-K210命令行开发环境搭建指南》

- 1) 下载示例程序[baidu flower](#)，将 `baidu_flower` 目录放到kendryte-standalone-sdk的src目录下
- 2) 下载在AI Studio上训练好的模型mobilenet.kmodel到 `baidu_flower` 目录下
- 3) 在[kendryte-standalone-sdk](#)（请使用develop分支的代码）目录新建build文件夹。如：
D:\Documents\kendryte-standalone-sdk\build
- 4) 编译

```
#打开cmd进入build目录
cd D:\Documents\kendryte-standalone-sdk\build
cmake .. -DPROJ=baidu_flower -G "MinGW Makefiles"
make
```

编译完成在build目录生成baidu_flower.bin

5) 烧录程序

K-flash.exe安装参考文档《PaddlePi-K210命令行开发环境搭建指南》

开发板插上USB Type-C，上电进入ISP模式。运行K-flash.exe，选择 Device、设置波特率、选择 baidu_flower.bin，点击 Flash 开始下载。

6) 运行测试

烧录完成后自动重启，使用图片进行测试（可使用AI Studio里面的数据集图片进行测试）。

ubuntu环境

交叉编译环境和SDK 参考文档《PaddlePi-K210命令行开发环境搭建指南》

1) 下载示例程序**baidu_flower**，将 baidu_flower 目录放到kendryte-standalone-sdk的src目录下，如：/home/tienfeek/Documents/kendryte/kendryte-standalone-sdk/src/baidu_flower

2) 下载在AI Studio上训练好的模型mobilenet.kmodel到 baidu_flower 目录下，如：/home/tienfeek/Documents/kendryte/kendryte-standalone-sdk/src/baidu_flower/mobilenet.kmodel

3) 在**kendryte-standalone-sdk**（请使用develop分支的代码）目录新建build文件夹。如：/home/tienfeek/Documents/kendryte/kendryte-standalone-sdk/build

4) 编译

```
#打开terminal进入build目录
cd /home/tienfeek/Documents/kendryte/kendryte-standalone-sdk/build
cmake .. -DPROJ=baidu_flower -DTOOLCHAIN=/opt/kendryte-toolchain/bin
make
```

编译完成在build目录生成baidu_flower.bin

5) 烧录应用

kflash.py安装脚本参考文档《PaddlePi-K210命令行开发环境搭建指南》

用USB Type-C线连接PaddlePi-K210 和开发电脑。执行如下命令将baidu_flower.bin 烧录到PaddlePi-K210开发板。

```
#进入到kflash.py所在目录
sudo python3 kflash.py -p /dev/ttyUSB0
/home/tienfeek/Documents/kendryte/kendryte-standalone-sdk/build/baidu_flower.bin
```

6) 运行测试

烧录完成后重启，使用图片进行测试（可使用AI Studio里面的数据集图片进行测试）。

2.2.2 PaddlePi_tiny-yolo编译和烧录程序

windows环境

交叉编译环境和SDK 参考文档《PaddlePi-K210命令行开发环境搭建指南》

1) 下载示例程序[baidu_screw](#)，将 `baidu_screw` 目录放到kendryte-standalone-sdk的src目录下

2) 下载在AI Studio上训练好的模型tiny_yolo.kmodel到 `baidu_screw` 目录下

3) 在[kendryte-standalone-sdk](#) (请使用develop分支的代码) 目录新建build文件夹。如：
D:\Documents\kendryte-standalone-sdk\build

4) 编译

```
#打开cmd进入build目录
cd D:\Documents\kendryte-standalone-sdk\build
cmake .. -DPROJ=baidu_screw -G "MinGW Makefiles"
make
```

编译完成在build目录生成baidu_screw.bin

5) 烧录程序

K-flash.exe安装参考文档《PaddlePi-K210命令行开发环境搭建指南》

开发板插上USB Type-C，上电进入ISP模式。运行K-flash.exe，选择 Device、设置波特率、选择 baidu_screw.bin，点击 Flash 开始下载。

6) 运行测试

烧录完成后自动重启，使用图片进行测试（可使用AI Studio里面的数据集图片进行测试）。

ubuntu环境

交叉编译环境和SDK 参考文档《PaddlePi-K210命令行开发环境搭建指南》

1) 下载示例程序[baidu_screw](#)，将 `baidu_screw` 目录放到kendryte-standalone-sdk的src目录下。如：/home/tienfeek/Documents/kendryte/kendryte-standalone-sdk/src/baidu_screw

2) 下载在AI Studio上训练好的模型tiny_yolo.kmodel到 `baidu_screw` 目录下，
如：/home/tienfeek/Documents/kendryte/kendryte-standalone-sdk/src/baidu_screw/tiny_yolo.kmodel

3) 在[kendryte-standalone-sdk](#) (请使用develop分支的代码) 目录新建build文件夹。
如：/home/tienfeek/Documents/kendryte/kendryte-standalone-sdk/build

4) 编译

```
#打开terminal进入build目录
cd /home/tienfeek/Documents/kendryte/kendryte-standalone-sdk/build
cmake .. -DPROJ=baidu_screw -DTOOLCHAIN=/opt/kendryte-toolchain/bin
make
```

编译完成在build目录生成baidu_screw.bin

5) 烧录应用

kflash.py安装脚本参考文档《PaddlePi-K210命令行开发环境搭建指南》

用USB Type-C线连接PaddlePi-K210 和开发电脑。执行如下命令将baidu_screw.bin 烧录到PaddlePi-K210开发板。

```
#进入到kflash.py所在目录  
sudo python3 kflash.py -p /dev/ttyUSB0  
/home/tienfeek/Documents/kendryte/kendryte-standalone-sdk/build/baidu_screw.bin
```

6) 运行测试

烧录完成后重启，使用图片进行测试（可使用AI Studio里面的数据集图片进行测试）。