

传输模型文件至TF卡

传输模型文件至TF卡

- 1、模型文件说明
- 2、导入模型文件
- 3、运行程序

1、模型文件说明

模型文件通常包含了神经网络架构的参数和权重信息，是神经网络训练过程的结果。对于K210而言，模型可以用于人脸检测、口罩识别、人脸识别、物体识别等各种机器学习任务。由于K210芯片资源有限，所以训练好的模型需要传输至TF卡储存，模型文件的后缀为.kmodel

2、导入模型文件

传输模型文件至TF卡的方式有两种，方式一是使用读卡器，将TF卡插入读卡器，读卡器插入电脑USB口，下载资料中的模型文件压缩包并解压，将KPU文件夹复制到H:盘的根目录。

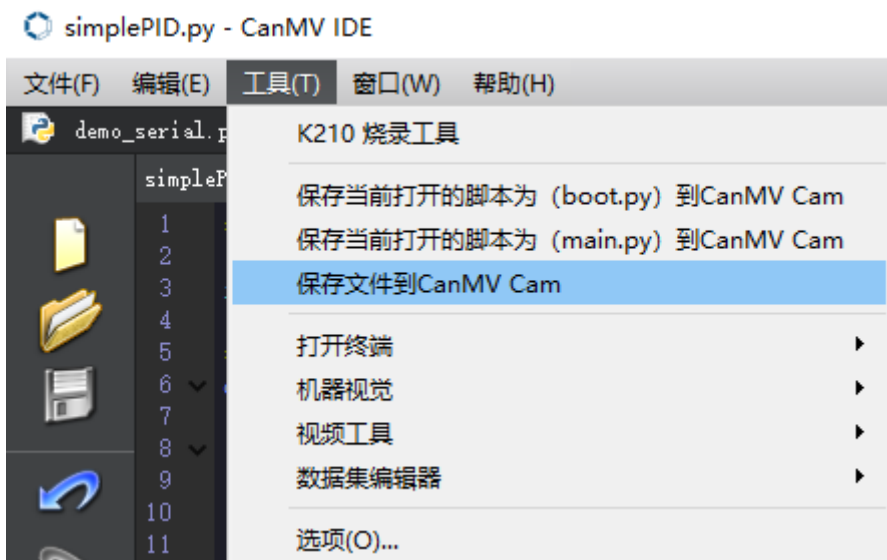
K210_SDCARD (H:)		
	名称	修改日期
	KPU	2023/1/11 15:17
	main.py	2023/2/22 9:58

方式一可以一次导入所有模型文件，省去每次导入模型文件的操作，以不同路径来区分加载模型文件。所有模型文件都放到KPU文件夹下，请不要修改KPU以及KPU目录以下的文件名称，否则默认例程程序会出现找不到模型文件的问题。

方式二是通过CanMV IDE来导入模型文件，无需借助读卡器，这里以导入landmark68.kmodel为例，将K210模块通过microUSB数据线连接到电脑USB口。然后点击CanMV IDE左下角连接按钮，连接成功会将图标变为已连接状态，同时灰色播放键变为绿色。

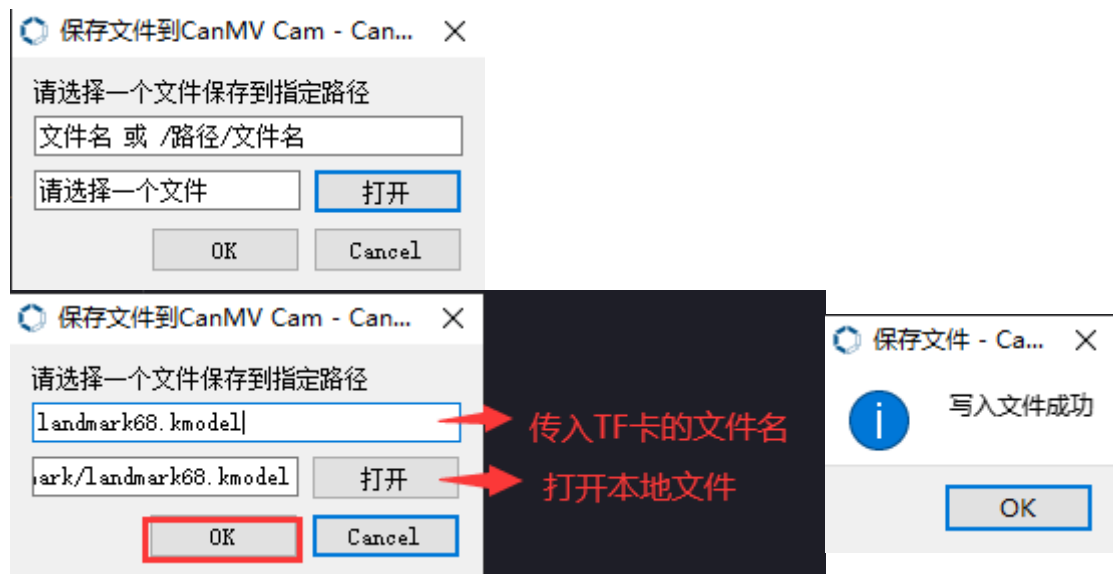


打开工具->保存文件到CanMV Cam。



点击打开，然后选择要传入的模型文件，传入内存卡。

注意：这里写入的文件优先写入到内存卡，保存时请删掉文件路径，带路径保存容易报错。



3、运行程序

运行代码时，以具体保存的模型路径为准，如按照本教程方式一操作，则使用默认代码，无需修改。

face_detect_68lm.py - CanMV IDE

文件(F) 编辑(E) 工具(T) 窗口(W) 帮助(H)

face_detect_68lm.py

```
1 import sensor, image, time, lcd
2 from maix import KPU
3
4
5 lcd.init()
6 sensor.reset()
7 sensor.set_pixformat(sensor.RGB565)
8 sensor.set_framesize(sensor.QVGA)
9 sensor.skip_frames(time = 100)
10 clock = time.clock()
11
12 anchor = (0.1075, 0.126875, 0.126875, 0.175, 0.1465625, 0.2246875, 0.1953125, 0.25375, 0.2440625, 0.2440625)
13 kpu = KPU()
14 kpu.load_kmodel("/sd/KPU/yolo face detect/face detect 320x240.kmodel")
15 kpu.init_yolo2(anchor, anchor_num=9, img_w=320, img_h=240, net_w=320, net_h=240, layer_w=10, layer_h=10)
16
17 lm68_kpu = KPU()
18 print("ready load model")
19 lm68_kpu.load_kmodel("/sd/KPU/face_detect_with_68landmark/landmark68.kmodel")
20
21
22 def extend_box(x, y, w, h, scale):
23     x1_t = x - scale*w
24     x2_t = x + w + scale*w
25     y1_t = y - scale*h
26     y2_t = y + h + scale*h
27     x1 = int(x1_t) if x1_t>1 else 1
28     x2 = int(x2_t) if x2_t<320 else 319
29     y1 = int(y1_t) if y1_t>1 else 1
30     y2 = int(y2_t) if y2_t<240 else 239
31     cut_img_w = x2-x1+1
32     cut_img_h = y2-y1+1
33     return x1, y1, cut_img_w, cut_img_h
```

如按照本教程的方式二操作，需要将程序中的多余路径删掉，即仅保留前缀/sd/。

face_detect_68lm.py*

face_detect_68lm.py*

```
1 import sensor, image, time, lcd
2 from maix import KPU
3
4
5 lcd.init()
6 sensor.reset()
7 sensor.set_pixformat(sensor.RGB565)
8 sensor.set_framesize(sensor.QVGA)
9 sensor.skip_frames(time = 100)
10 clock = time.clock()
11
12 anchor = (0.1075, 0.126875, 0.126875, 0.175, 0.1465625, 0.2246875, 0.1953125, 0.25375, 0.2440625, 0.2440625)
13 kpu = KPU()
14 kpu.load_kmodel("/sd/face_detect_320x240.kmodel")
15 kpu.init_yolo2(anchor, anchor_num=9, img_w=320, img_h=240, net_w=320, net_h=240, layer_w=10, layer_h=10)
16
17 lm68_kpu = KPU()
18 print("ready load model")
19 lm68_kpu.load_kmodel("/sd/landmark68.kmodel")
20
21
22 def extend_box(x, y, w, h, scale):
23     x1_t = x - scale*w
24     x2_t = x + w + scale*w
25     y1_t = y - scale*h
26     y2_t = y + h + scale*h
27     x1 = int(x1_t) if x1_t>1 else 1
28     x2 = int(x2_t) if x2_t<320 else 319
29     y1 = int(y1_t) if y1_t>1 else 1
30     y2 = int(y2_t) if y2_t<240 else 239
31     cut_img_w = x2-x1+1
32     cut_img_h = y2-y1+1
33     return x1, y1, cut_img_w, cut_img_h
```