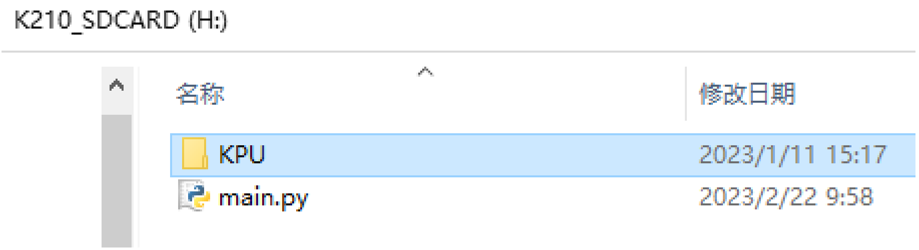
Transfer model files to TF card

1、Model file description

The model file usually contains the parameters and weight information of the neural network architecture and is the result of the neural network training process. For K210, the model can be used for various machine learning tasks such as face detection, mask recognition, face recognition, object recognition, etc. Due to the limited resources of the K210 chip, the trained model needs to be transferred to the TF card for storage. The suffix of the model file is .kmodel.

2、Import model files

There are two ways to transfer the model file to the TF card. The first way is to use a card reader. Insert the TF card into the card reader. The card reader is inserted into the USB port of the computer. Download the compressed model file package in the data and decompress it. Copy the folder to the root directory of the H: drive.

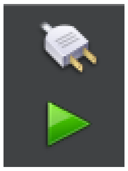


K210_SDCARD (H:)

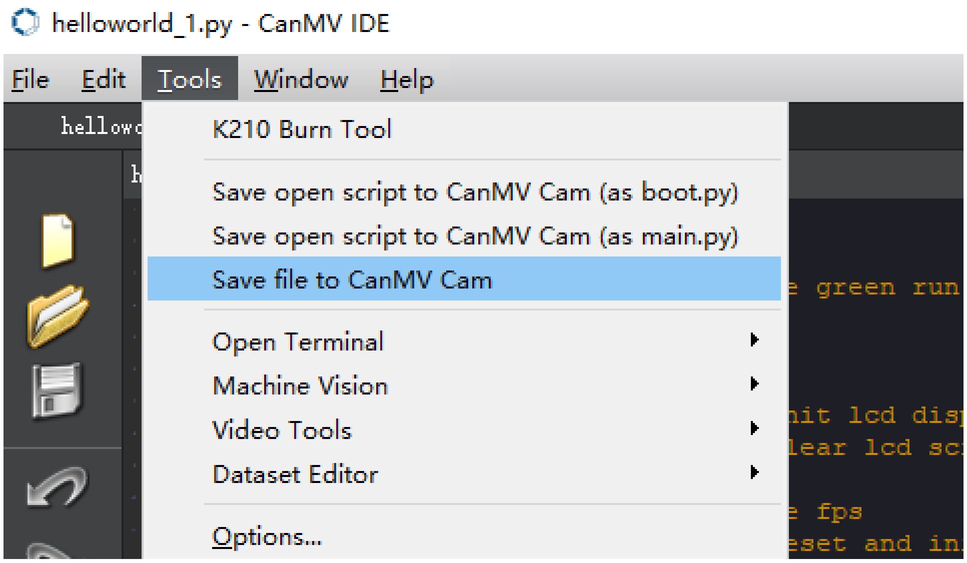
名称	修改日期
KPU	2023/1/11 15:17
main.py	2023/2/22 9:58

Method 1 can import all model files at once, eliminating the need to import model files each time, and load model files using different paths. All model files are placed in the KPU folder. Please do not modify KPU and the file names under the KPU directory, otherwise the default routine program will not be able to find the model files.

The second method is to import the model file through the CanMV IDE software without using a card reader. Here we take the import of landmark68.kmodel as an example and connect the K210 module to the computer USB port through a microUSB data cable. Then click the connect button in the lower left corner of CanMV IDE. If the connection is successful, the icon will change to the connected state, and the gray play button will turn green.

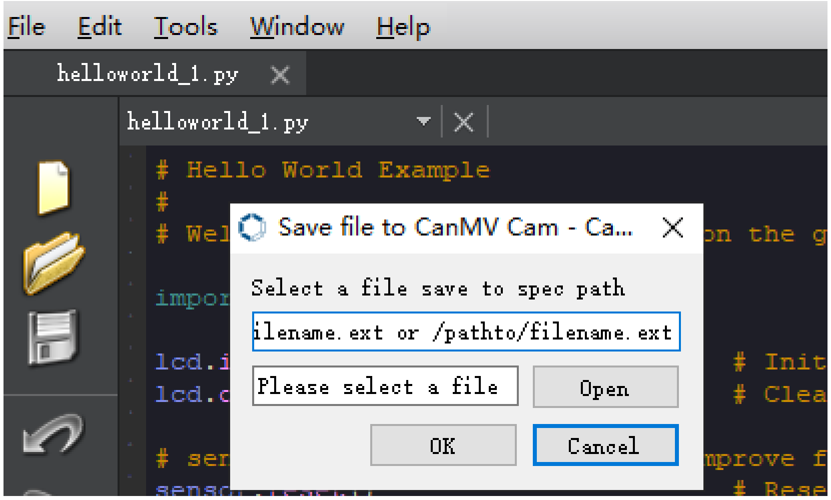


Open Tools->Save file to CanMV Cam.



Click Open, then select the model file to be imported and transfer it to the memory card.

Note: The files written here are written to the memory card first. Please delete the file path when saving. Saving with a path is likely to result in an error.

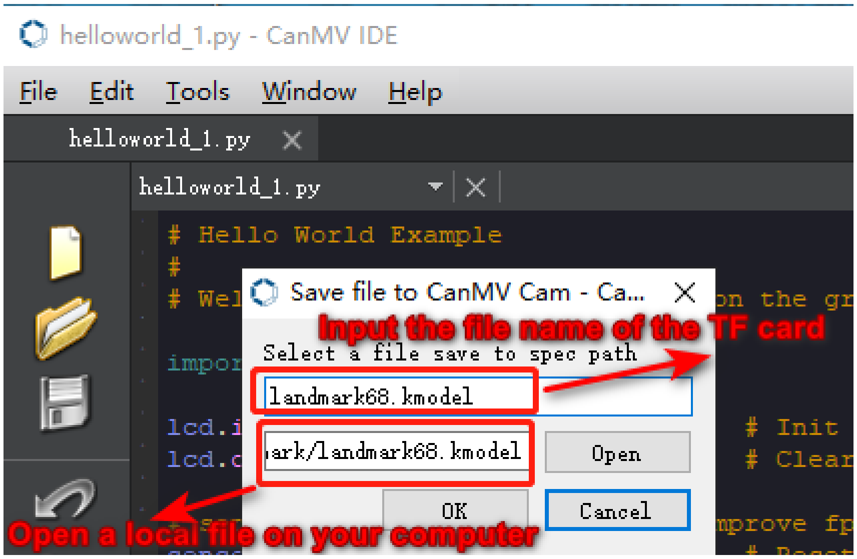


Select a file save to spec path

ilename.ext or /pathto/filename.ext

Please select a file

```
# Hello World Example
#
# Wel
impor
lcd.i
lcd.c
# ser
```



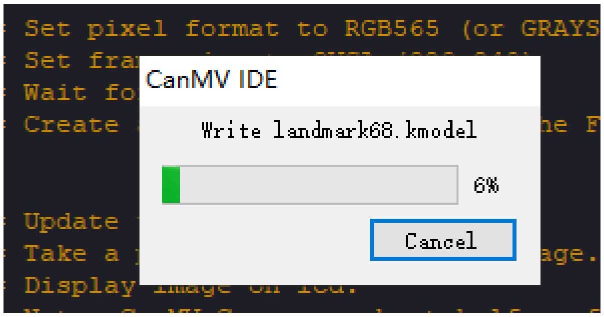
Input the file name of the TF card

Select a file save to spec path

landmark68.kmodel

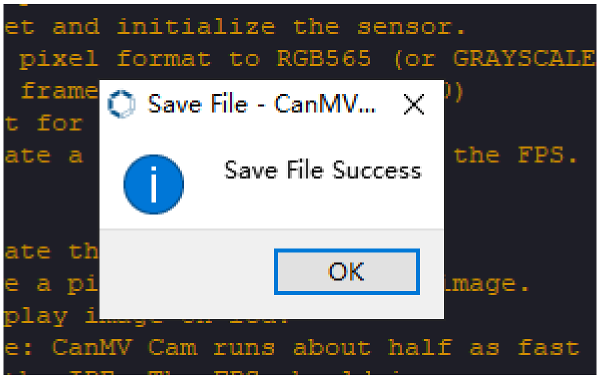
ark/landmark68.kmodel

Open a local file on your computer



Write landmark68.kmodel

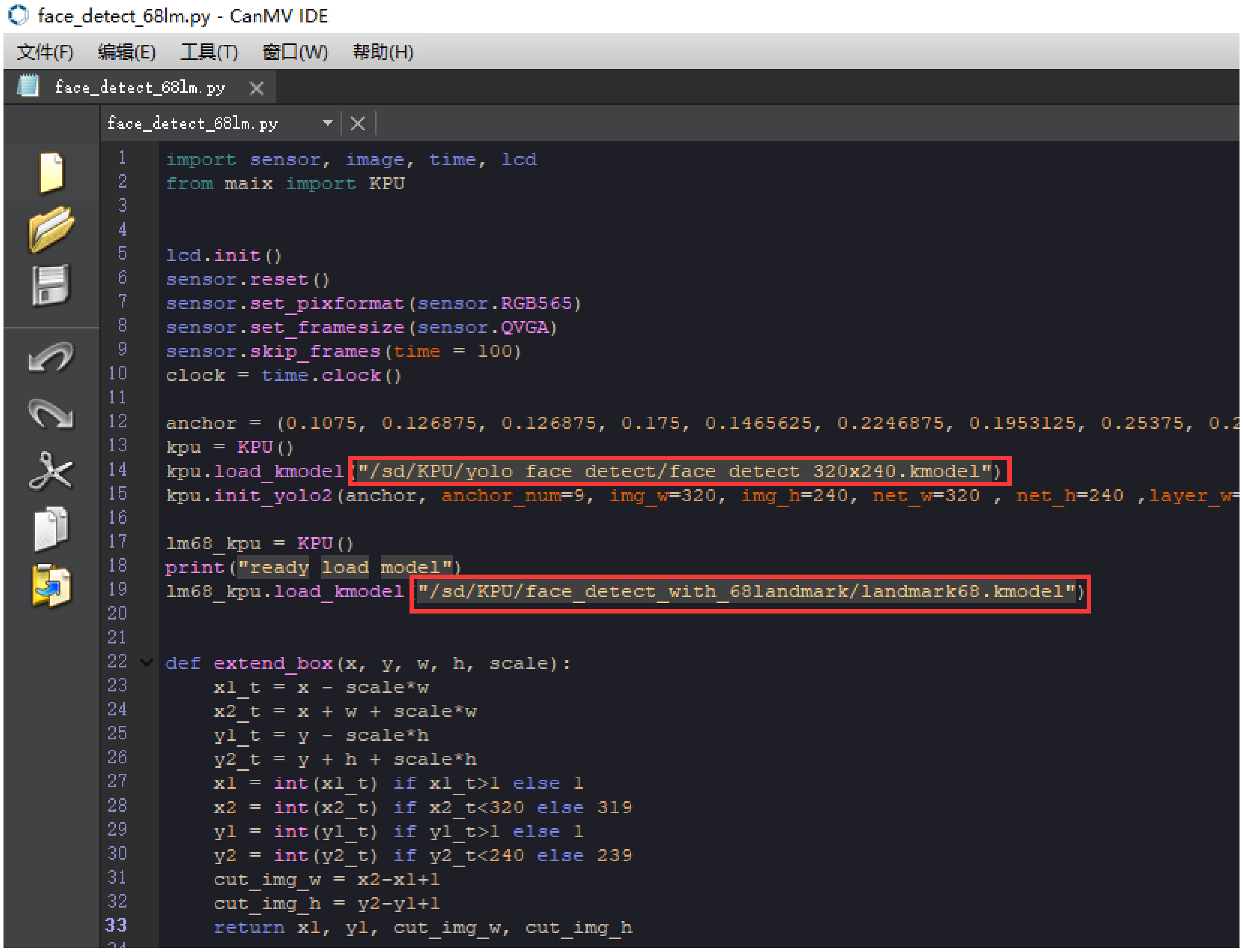
6%



Save File Success

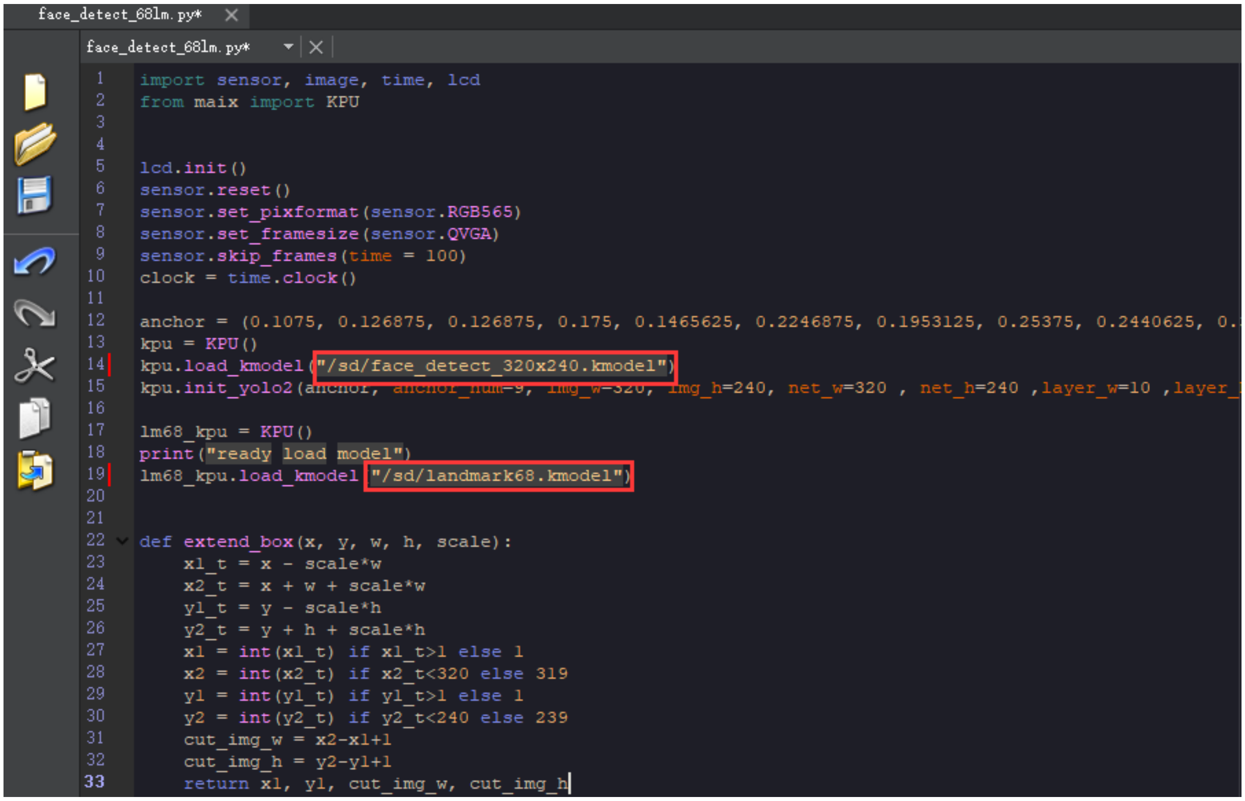
3、Run program

when running code，The specific saved model path shall prevail. If you follow the first method of this tutorial, the default code will be used without modification.



```
import sensor, image, time, lcd
from maix import KPU


lcd.init()
sensor.reset()
sensor.set_pixformat(sensor.RGB565)
sensor.set_framesize(sensor.QVGA)
sensor.skip_frames(time = 100)
clock = time.clock()

anchor = (0.1075, 0.126875, 0.126875, 0.175, 0.1465625, 0.2246875, 0.1953125, 0.25375, 0.2
kpu = KPU()
kpu.load_kmodel("/sd/KPU/yolo_face_detect/face_detect_320x240.kmodel")
kpu.init_yolo2(anchor, anchor_num=9, img_w=320, img_h=240, net_w=320 , net_h=240 ,layer_w=

lm68_kpu = KPU()
print("ready load model")
lm68_kpu.load_kmodel("/sd/KPU/face_detect_with_68landmark/landmark68.kmodel")


def extend_box(x, y, w, h, scale):
    x1_t = x - scale*w
    x2_t = x + w + scale*w
    y1_t = y - scale*h
    y2_t = y + h + scale*h
    x1 = int(x1_t) if x1_t>1 else 1
    x2 = int(x2_t) if x2_t<320 else 319
    y1 = int(y1_t) if y1_t>1 else 1
    y2 = int(y2_t) if y2_t<240 else 239
    cut_img_w = x2-x1+1
    cut_img_h = y2-y1+1
    return x1, y1, cut_img_w, cut_img_h
```

If you follow the second method of this tutorial, you need to delete the redundant paths in the program, that is, only keep the prefix **/sd/**.



```
import sensor, image, time, lcd
from maix import KPU


lcd.init()
sensor.reset()
sensor.set_pixformat(sensor.RGB565)
sensor.set_framesize(sensor.QVGA)
sensor.skip_frames(time = 100)
clock = time.clock()

anchor = (0.1075, 0.126875, 0.126875, 0.175, 0.1465625, 0.2246875, 0.1953125, 0.25375, 0.2440625, 0.
kpu = KPU()
kpu.load_kmodel("/sd/face_detect_320x240.kmodel")
kpu.init_yolo2(anchor, anchor_num=9, img_w=320, img_h=240, net_w=320 , net_h=240 ,layer_w=10 ,layer_

lm68_kpu = KPU()
print("ready load model")
lm68_kpu.load_kmodel("/sd/landmark68.kmodel")


def extend_box(x, y, w, h, scale):
    x1_t = x - scale*w
    x2_t = x + w + scale*w
    y1_t = y - scale*h
    y2_t = y + h + scale*h
    x1 = int(x1_t) if x1_t>1 else 1
    x2 = int(x2_t) if x2_t<320 else 319
    y1 = int(y1_t) if y1_t>1 else 1
    y2 = int(y2_t) if y2_t<240 else 239
    cut_img_w = x2-x1+1
    cut_img_h = y2-y1+1
    return x1, y1, cut_img_w, cut_img_h
```