

15. License plate recognition

15. License plate recognition

- 15.1 Experimental objectives
- 15.2 Preparation before the experiment
- 15.3 Experimental process
- 15.4 Experimental results
- 15.5 Experimental summary

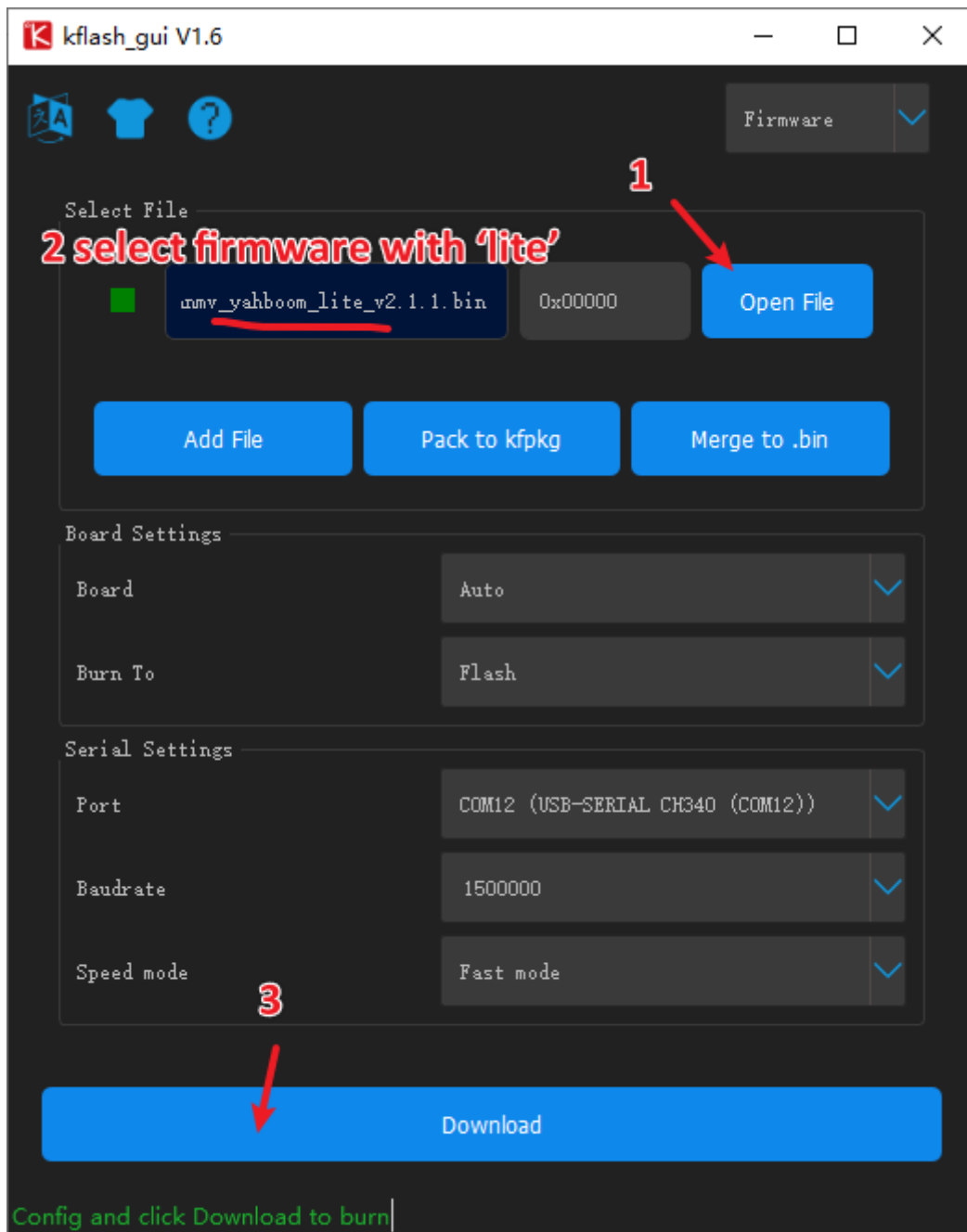
15.1 Experimental objectives

This lesson mainly studies the K210 license plate recognition function.

The reference code path for this experiment is: CanMV\05-AI\car_licenseplate_recog_cn.py

15.2 Preparation before the experiment

! Note! : Since the model and weight files used in this case are large, the lite version firmware must be flashed before running, otherwise it will freeze.



Please confirm whether the sd card model, font library and weight file are imported successfully:

1. /sd/KPU/car_licenseplate_recog/lp_detect.kmodel
2. /sd/KPU/car_licenseplate_recog/lp_recog.kmodel
3. /sd/0xA00000.Dzk
4. /sd/KPU/car_licenseplate_recog/lp_weight.bin

15.3 Experimental process

The factory firmware of the module has integrated the AI vision algorithm module. If you have downloaded other firmware, please burn it back to the factory firmware before conducting the experiment.

1. Initialize the font library and the parameters required for the model

```

image.font_load(image.UTF8, 16, 16, "/sd/0xA00000.Dzk")
#image.font_load(image.UTF8, 16, 16, 0xA00000)

province = ("皖沪津渝冀晋蒙辽吉黑苏浙京闽赣鲁豫鄂湘粤桂琼川贵云藏陕甘青宁新")
ads = ('A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'J', 'K', 'L', 'M', 'N', 'P',
'Q', 'R', 'S', 'T', 'U', 'V', 'W', 'X', 'Y', 'Z', '0', '1', '2', '3', '4', '5',
'6', '7', '8', '9')
names = []
anchor = (8.30891522166988, 2.75630994889035, 5.18609903718768,
1.7863757404970702, 6.91480529053198, 3.825771881004435, 10.218567655549439,
3.69476690620971, 6.4088204258368195, 2.38813526350986)

```

2. Loading model file

```

lp_det_kpu = KPU()
lp_det_kpu.load_kmodel("/sd/KPU/car_licenseplate_recog/lp_detect.kmodel")
lp_det_kpu.init_yolo2(anchor, anchor_num=len(anchor) // 2, img_w=320, img_h=240,
net_w=320, net_h=240, layer_w=20, layer_h=15, threshold=0.7, nms_value=0.3,
classes=len(names))

lp_recog_kpu = KPU()
lp_recog_kpu.load_kmodel("/sd/KPU/car_licenseplate_recog/lp_recog.kmodel")

# load after-process data
lp_recog_kpu.lp_recog_load_weight_data("/sd/KPU/car_licenseplate_recog/lp_weight
.bin")

```

3. Define the function for drawing the license plate border

```

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<256 else 255
    cut_img_w = x2-x1+1
    cut_img_h = y2-y1+1
    return x1, y1, cut_img_w, cut_img_h

```

4. Main program: frame the license plate and recognize the license plate text

```

try:
    while 1:
        gc.collect()
        clock.tick() # Update the FPS clock.
        img = sensor.snapshot()
        lp_det_kpu.run_with_output(img)
        lps = lp_det_kpu.regionlayer_yolo2()
        for lp in lps:
            # 框出车牌位置 Frame the license plate location
            x, y, w, h = extend_box(lp[0], lp[1], lp[2], lp[3], 0.08)
            img.draw_rectangle(x, y, w, h, color=(0, 255, 0))

```

```

# 识别车牌 License plate recognition
lp = []
lp_img = img.cut(x, y, w, h)
resize_img = lp_img.resize(208, 64)
resize_img.replace(hmirror=True)
resize_img.pix_to_ai()
lp_recog_kpu.run_with_output(resize_img)
output = lp_recog_kpu.lp_recog()
for o in output:
    lp.append(o.index(max(o)))
show_lp_str = "%s %s-%s%s%s%s" %(province[lp[0]],
ads[lp[1]],ads[lp[2]], ads[lp[3]], ads[lp[4]], ads[lp[5]], ads[lp[6]])
# 此处需要重新加载字体文件，否则中文显示会异常
# You need to reload the font file here, otherwise the Chinese
display will be abnormal
image.font_load(image.UTF8, 16, 16, "/sd/0xA00000.Dzk")
img.draw_string(x + 2, y - 20 - 2, "%s"%show_lp_str,color=(255, 0,
0),x_spacing=2, mono_space=False,scale=2)
del lp
del lp_img
del resize_img
gc.collect()
lcd.display(img)
except Exception as e:
    print(e)
    lp_det_kpu.deinit()
    lp_recog_kpu.deinit()
del lp_det_kpu
del lp_recog_kpu
gc.collect()

```

15.4 Experimental results

Connect the K210 module to the computer via a microUSB data cable, click the connection button in CanMV IDE, and click the run button after the connection is completed to run the example code. You can also download the code as main.py to the K210 module and run it.

After waiting for the system to initialize, the LCD displays the camera image, and the license plate on the screen will be framed by a green frame and the license plate information will be recognized.



15.5 Experimental summary

Hand recognition requires a memory card to load the model file, so you need to import the model file into the memory card in advance, and then insert the memory card into the memory card slot of the K210 module. If the model file in the memory card cannot be read, an error will be reported. The model and weight files used in this case are large, and the lite version firmware needs to be flashed before running, otherwise it will freeze. In this experiment, we learned how to implement the license plate recognition function based on yolo on the K210 platform. Due to the limitation of model size, the recognition accuracy may be affected. You can train a better model by yourself.