

K210

安装

使用YOLO训练的网络进行识别

[在线训练](#)

[本地训练](#)

[效果](#)

[串口通信](#)

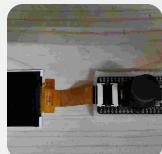
[PWM输出](#)

安装

AI嵌入式K210项目（19）—安装CanMV IDE开发软件_canmvide—CSDN博客

文章浏览阅读4.5k次，点赞37次，收藏43次。前几章我们介绍K210使用C语言裸机开发方法，大家对K...

https://blog.csdn.net/bin_zhang1/article/details/135656400?ops_request_misc=%257B%2522re...



K210学习笔记（二）——学习前准备_kflash gui—CSDN博客

文章浏览阅读1.2w次，点赞22次，收藏137次。一、开发板的选择我现在手上有两块K210开发板，一块...

https://blog.csdn.net/Thousand_drive/article/details/123822939?ops_request_misc=%257B%2522re...

【【正点原子】手把手教你学DNK210快速入门视频教程】



第8讲 CanMV简介及CanMV IDE安装_哔哩哔哩_bilibili

第8讲 CanMV简介及CanMV IDE安装是【正点原子】手把手教你学DNK210快速入门视频教程的第8集...

https://www.bilibili.com/video/BV1kD421G7fu?p=8&vd_source=47dedd4e022eac07e87bc4dc59...



ATK-DNK210开发板 — 正点原子资料下载中心 1.0.0 文档

<http://www.openedv.com/docs/boards/k210/ATK-DNK210.html>

软件地址 [Releases · kendryte/canmv_ide](#)

LCD显示问题



K210显示画面_c# 连接k210 显示帧-CSDN博客

文章浏览阅读1k次，点赞23次，收藏23次。本文详细介绍了如何在MaixPyIDE中初始化单目摄像头、...

https://blog.csdn.net/qq_64366975/article/details/135723047?ops_request_misc=&request_id=...

使用YOLO训练的网络进行识别



【K210】 K210学习笔记六——MaixHub在线模型训练识别数字_k210识别激光-C...

文章浏览阅读1.4w次，点赞81次，收藏490次。本文着重于如何使用MaixHub平台，在线训练模型，识...

https://blog.csdn.net/adas323/article/details/126534658?ops_request_misc=%257B%2522req...

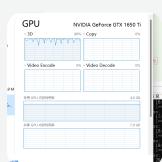
在线训练

开源数据集！ Roboflow: Computer vision tools for developers and enterprises

训练网站！ MaixHub, 嘉楠开发者社区

本地训练

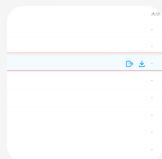
mx-yolov3



Mx_yolov3的安装并使用GPU训练_mx yolov3 3.0下载-CSDN博客

文章浏览阅读3.6k次。小白为了使用GPU训练模型，从各位大佬那里找来的解决方法，希望可以给大家...

https://blog.csdn.net/m0_67565756/article/details/125809183?ops_request_misc=%257B%2522...



手把手教你部署K210本地Mx_yolov3环境（最新）_mx-yolov3环境-CSDN博客

文章浏览阅读6.6k次，点赞36次，收藏188次。本地部署K210的yolov2环境_mx-yolov3环境

https://blog.csdn.net/m0_73841621/article/details/131855356?ops_request_misc=&request_id=...



Mx-yolov3+Maixpy+ K210进行本地模型训练和目标检测-CSDN博客

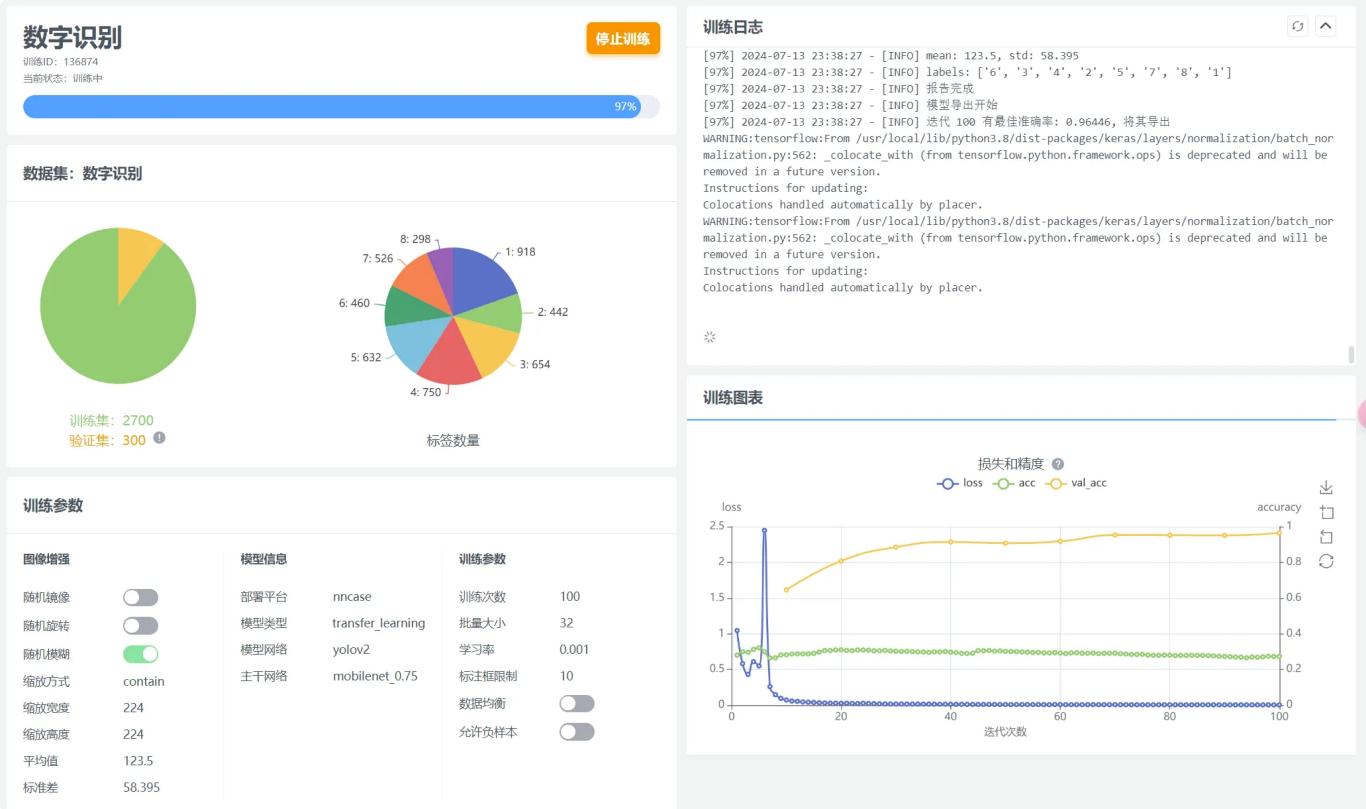
文章浏览阅读2.4w次，点赞88次，收藏564次。我接触K210同样也是因为一次比赛，需要进行目标检...

https://blog.csdn.net/qq_51963216/article/details/121044449?ops_request_misc=&request_id=...





效果



串行端口 - COM25 - 115,200 BPS - MaixPy IDE

串行终端

```
({"x":115, "y":0, "w":57, "h":75, "value":0.959240, "classid":0, "index":2, "objnum":4}
{"x":122, "y":123, "w":56, "h":77, "value":0.793928, "classid":6, "index":3, "objnum":4}
{"x":28, "y":91, "w":80, "h":123, "value":0.97123, "classid":5, "index":0, "objnum":4}
{"x":115, "y":95, "w":80, "h":123, "value":0.697024, "classid":6, "index":1, "objnum":4}
{"x":42, "y":0, "w":56, "h":73, "value":0.957417, "classid":4, "index":2, "objnum":4}
{"x":122, "y":0, "w":56, "h":71, "value":0.932494, "classid":0, "index":3, "objnum":4}
{"x":30, "y":91, "w":80, "h":123, "value":0.982371, "classid":5, "index":0, "objnum":4}
{"x":115, "y":95, "w":80, "h":123, "value":0.657577, "classid":6, "index":1, "objnum":4}
{"x":45, "y":0, "w":56, "h":73, "value":0.927125, "classid":4, "index":2, "objnum":4}
{"x":30, "y":94, "w":80, "h":122, "value":0.972750, "classid":5, "index":0, "objnum":4}
{"x":115, "y":96, "w":80, "h":123, "value":0.571681, "classid":6, "index":1, "objnum":4}
{"x":45, "y":0, "w":56, "h":73, "value":0.896580, "classid":4, "index":2, "objnum":4}
{"x":125, "y":0, "w":56, "h":71, "value":0.811386, "classid":0, "index":3, "objnum":4}
{"x":33, "y":91, "w":80, "h":123, "value":0.988225, "classid":5, "index":0, "objnum":4}
{"x":126, "y":95, "w":80, "h":123, "value":0.528239, "classid":6, "index":1, "objnum":4}
{"x":40, "y":0, "w":56, "h":73, "value":0.954904, "classid":4, "index":2, "objnum":4}
{"x":125, "y":0, "w":56, "h":71, "value":0.810909, "classid":0, "index":3, "objnum":4}
{"x":30, "y":95, "w":80, "h":123, "value":0.929519, "classid":5, "index":0, "objnum":5}
{"x":126, "y":96, "w":80, "h":123, "value":0.583942, "classid":6, "index":1, "objnum":5}
{"x":124, "y":99, "w":80, "h":123, "value":0.528618, "classid":0, "index":2, "objnum":5}
{"x":48, "y":0, "w":56, "h":75, "value":0.970982, "classid":4, "index":3, "objnum":5}
{"x":125, "y":0, "w":56, "h":71, "value":0.867227, "classid":0, "index":4, "objnum":5}
{"x":125, "y":103, "w":80, "h":122, "value":0.904325, "classid":5, "index":0, "objnum":5}
{"x":48, "y":0, "w":56, "h":77, "value":0.983481, "classid":4, "index":1, "objnum":4}
{"x":135, "y":0, "w":56, "h":72, "value":0.888674, "classid":0, "index":2, "objnum":4}
{"x":129, "y":124, "w":56, "h":77, "value":0.600497, "classid":6, "index":3, "objnum":4}
{"x":30, "y":101, "w":80, "h":123, "value":0.904309, "classid":5, "index":0, "objnum":5}
{"x":123, "y":99, "w":80, "h":123, "value":0.602320, "classid":0, "index":1, "objnum":5}
{"x":48, "y":0, "w":56, "h":77, "value":0.985696, "classid":4, "index":2, "objnum":5}
{"x":135, "y":0, "w":56, "h":75, "value":0.886449, "classid":0, "index":3, "objnum":5}
 {"x":130, "y":123, "w":56, "h":77, "value":0.602170, "classid":6, "index":4, "objnum":5}
 {"x":28, "y":101, "w":80, "h":123, "value":0.936750, "classid":5, "index":0, "objnum":5}
 {"x":48, "y":0, "w":56, "h":77, "value":0.987801, "classid":4, "index":1, "objnum":4}
 {"x":125, "y":0, "w":56, "h":73, "value":0.866139, "classid":0, "index":2, "objnum":4}
 {"x":130, "y":123, "w":56, "h":77, "value":0.603560, "classid":6, "index":3, "objnum":4}
 {"x":28, "y":101, "w":80, "h":123, "value":0.936685, "classid":5, "index":0, "objnum":5}
 {"x":123, "y":99, "w":80, "h":123, "value":0.600081, "classid":0, "index":1, "objnum":5}
 {"x":48, "y":0, "w":56, "h":77, "value":0.987630, "classid":4, "index":2, "objnum":5}
 {"x":125, "y":0, "w":56, "h":72, "value":0.874574, "classid":0, "index":3, "objnum":5}
 {"x":130, "y":123, "w":56, "h":77, "value":0.784017, "classid":6, "index":4, "objnum":5}
 {"x":28, "y":101, "w":80, "h":123, "value":0.904335, "classid":5, "index":0, "objnum":4}
 {"x":48, "y":0, "w":56, "h":77, "value":0.986407, "classid":4, "index":1, "objnum":4}
 {"x":135, "y":0, "w":56, "h":72, "value":0.867901, "classid":0, "index":2, "objnum":4}
 {"x":130, "y":123, "w":56, "h":77, "value":0.700421, "classid":6, "index":3, "objnum":4}
 {"x":36, "y":68, "w":80, "h":123, "value":0.710043, "classid":5, "index":0, "objnum":1}
```

帧缓冲区

直方图

RGB色彩空间

Res - 没有图像

没有图像

平均数 0 中位数 0 众数 0 StDev 0

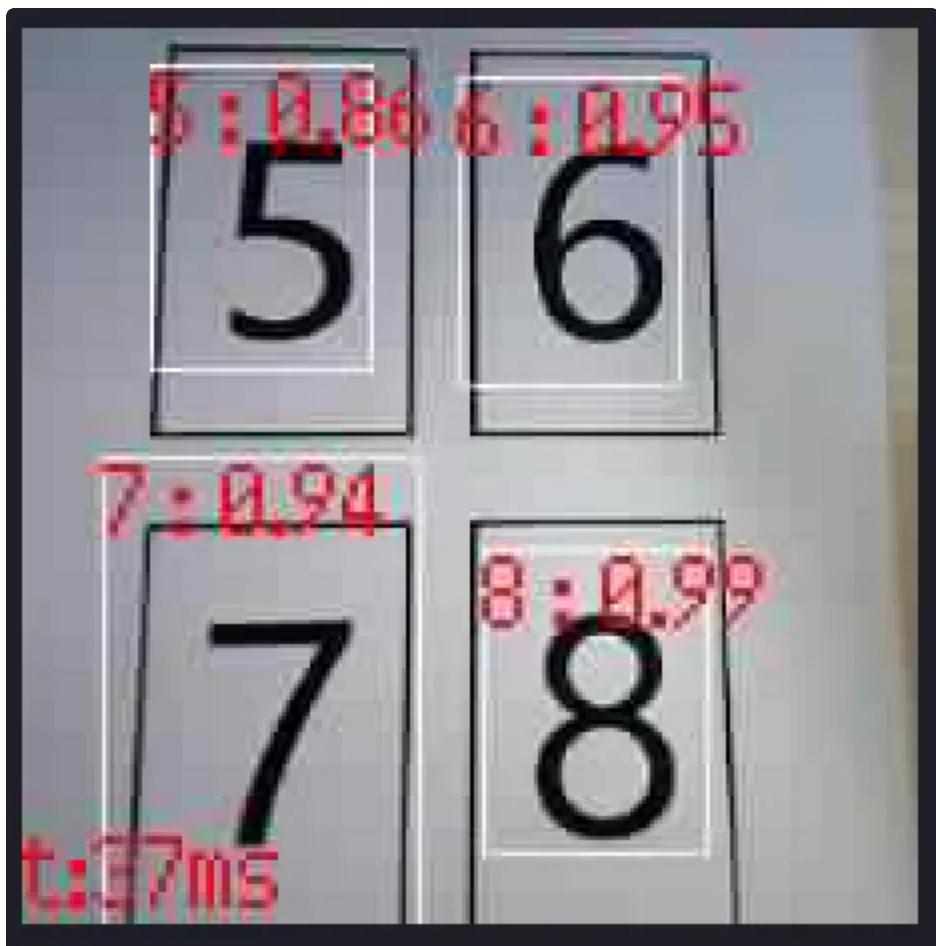
最小 0 最大 0 LQ 0 UQ 0

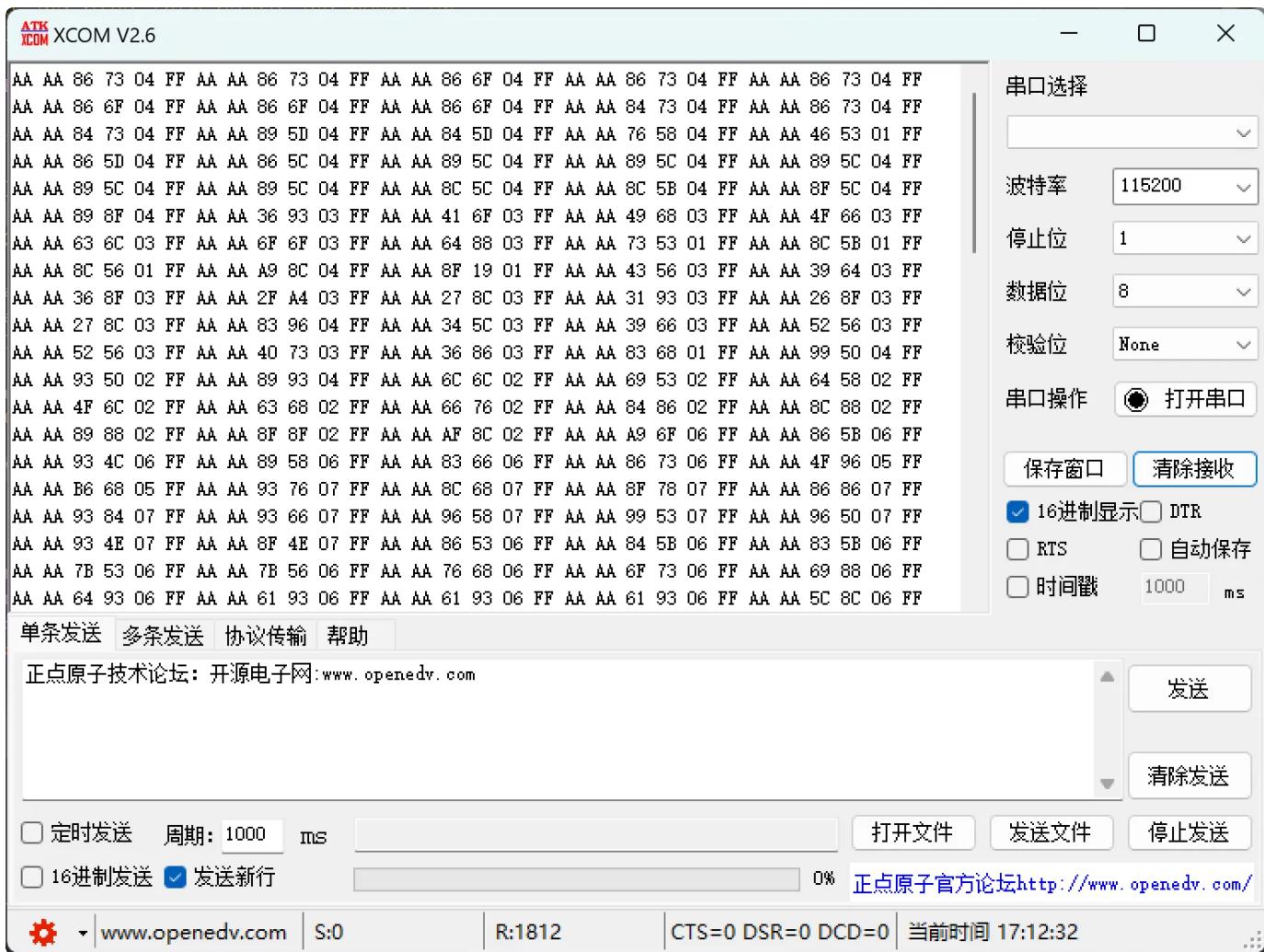
平均数 0 中位数 0 众数 0 StDev 0

最小 0 最大 0 LQ 0 UQ 0

平均数 0 中位数 0 众数 0 StDev 0

最小 0 最大 0 LQ 0 UQ 0





串口通信



STM32与K210串口通信_k210与stm32通信-CSDN博客

文章浏览阅读2.3w次，点赞114次，收藏823次。这篇文章是为了填上一篇埋下的坑，k210和STM32通...

https://blog.csdn.net/qq_70715693/article/details/130312626

▼ k210发送

Python |

```
1 # Untitled - By: wendy - 周一 7月 15 2024
2
3 import time
4 from machine import UART #串口库函数
5 from fpioa_manager import fm # GPIO重定向函数
6
7 fm.register(7, fm.fpioa.UART1_TX, force=True)
8 uart_A = UART(UART.UART1, 115200, 8, 0, 1, timeout=1000, read_buf_len=409
6)
9
10
11 def sending_data(x,y,z):
12     FH = bytearray([0x2C,0x12,x,y,z,0x5B])
13     uart_A.write(FH);
14
15 Cx = 0
16 Cy = 0
17 Cz = 0
18
19 while True:
20
21     Cx+=1;
22     Cy+=1;
23     Cz+=1;
24     sending_data(Cx,Cy,Cz)
25     print("Cx:",Cx,"Cy",Cy,"Cz:",Cz)
26     time.sleep_ms(1000)
27
```

▼ nuc环境配置

Bash |

```
1 pip install pyserial
```

```
1 import serial
2 import time
3
4 # 设置串口参数
5 serial_port = '/dev/ttyUSB0' # 根据实际情况修改串口号
6 baud_rate = 9600 # 波特率, 根据实际设置
7 ser = serial.Serial(serial_port, baud_rate)
8
9 try:
10     while True:
11         if ser.in_waiting > 0:
12             # 读取串口数据
13             data = ser.readline().decode('utf-8').strip()
14
15             # 处理读取到的数据
16             # 在这里可以添加你的数据处理逻辑, 比如打印、保存到文件等
17             print("Received:", data)
18
19             # 示例: 如果收到特定数据, 退出循环
20             if data == 'exit':
21                 break
22
23             # 可选: 添加其他后续处理逻辑
24
25             # 增加适当的延时, 以免过于频繁地读取串口数据
26             time.sleep(0.1)
27
28 finally:
29     # 关闭串口
30     ser.close()
```

PWM输出

Python |

```
1 # 初始化定时器和 PWM 用于舵机控制
2 tim1 = Timer(Timer.TIMER0, Timer.CHANNEL0, mode=Timer.MODE_PWM)
3 tim2 = Timer(Timer.TIMER0, Timer.CHANNEL1, mode=Timer.MODE_PWM)
4 pwm_x = PWM(tim1, freq=50, duty=7.5, pin=11, enable=True) # 水平方向舵机连接
5 # 到引脚 11
6 pwm_y = PWM(tim2, freq=50, duty=7.5, pin=12, enable=True) # 垂直方向舵机连接
7 # 到引脚 12
```

▼ 控制

Python |

```
1 # 水平 PID 控制
2 error_x = largest_blob.cx() - center_x
3 integral_x += error_x
4 derivative_x = error_x - previous_error_x
5 output_x = Kp * error_x + Ki * integral_x + Kd * derivative_x
6
7 # 控制水平方向舵机
8 current_duty_x = pwm_x.duty()
9 adjustment_x = min(max(output_x, -max_adjustment), max_adjustment)
10 new_duty_x = current_duty_x - adjustment_x
11 new_duty_x = max(min(new_duty_x, 12.5), 2.5)
12 pwm_x.duty(new_duty_x)
```