



西北大学
Northwest University

轻量化目标检测算法的设计

姓名: 薄劲阳 学号: 2020115025

专业: 计算机科学与技术

导师: 卜起荣

答辩日期: 2024 年 1 月 31 日

目录

CONTENTS

1

选题
背景

2

解决问题
与意义

3

技术
路线

4

工作
安排

选题背景

题目：轻量化目标检测算法设计

基于深度学习的目标检测算法在准确性方面表现出色，但它们通常具有较大的模型大小和高计算复杂度，导致在资源受限的环境下难以应用。因此通过牺牲一小部分检测精度而实现加速的轻量化网络获得了广泛关注。

目标检测算法现状：

模型参数多

高计算复杂度

难以部署到嵌入式边缘设备



解决问题与意义

在此背景下，利用Python语言和现有的Yolov5、Yolov5-Lite目标检测模型，设计轻量化的目标检测算法，并最终部署到树莓派。旨在降低目标检测模型的复杂度，进一步提高目标检测的速度。

模型训练的环境

名称	内容
操作系统	Ubuntu 22. 04. 3 LTS 64位
处理器	2. 5Ghz 双核
Python环境	3. 8版本、Pycharm
Python包及其他技术	pandas, sklearn, numpy, pytorch等
内存	16GB
显卡	NVIDIA GeForce RTX 4060

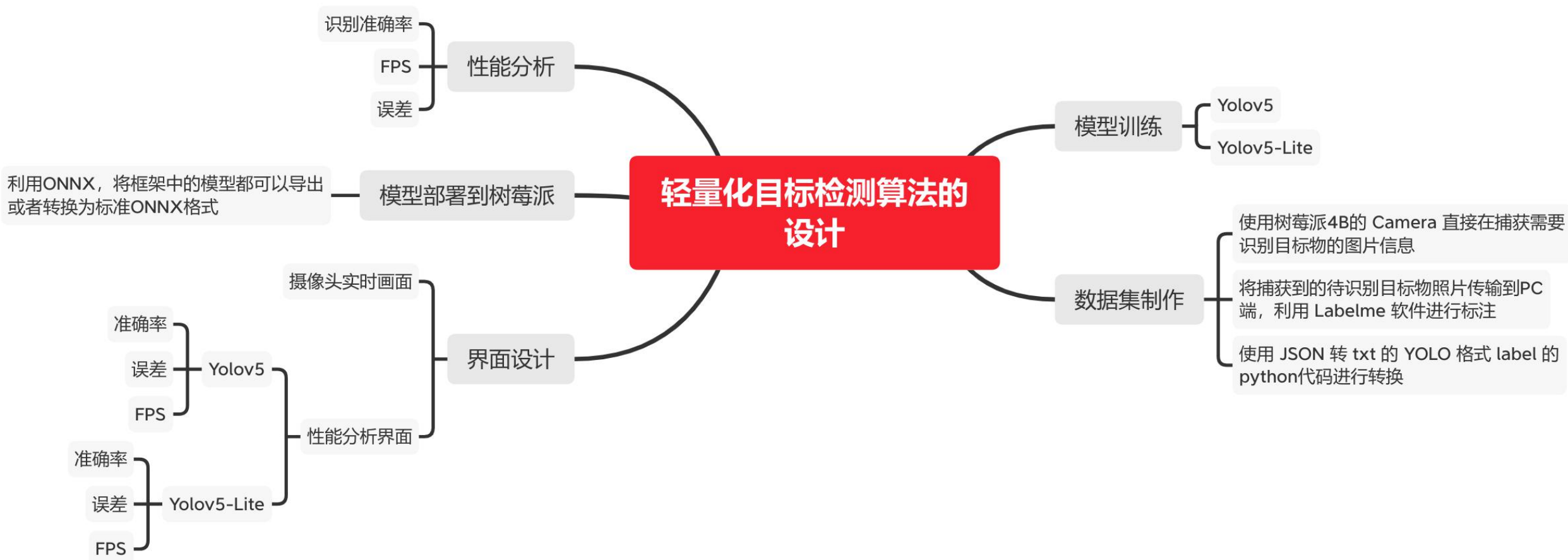
模型运行的环境

名称	内容
操作系统	Raspberry Pi OS
处理器	1. 5GHz四核
Python环境	3. 8版本
Python包及其他技术	pandas, sklearn, numpy, pytorch, Open CV, PyQt等
内存	4GB

技术路线

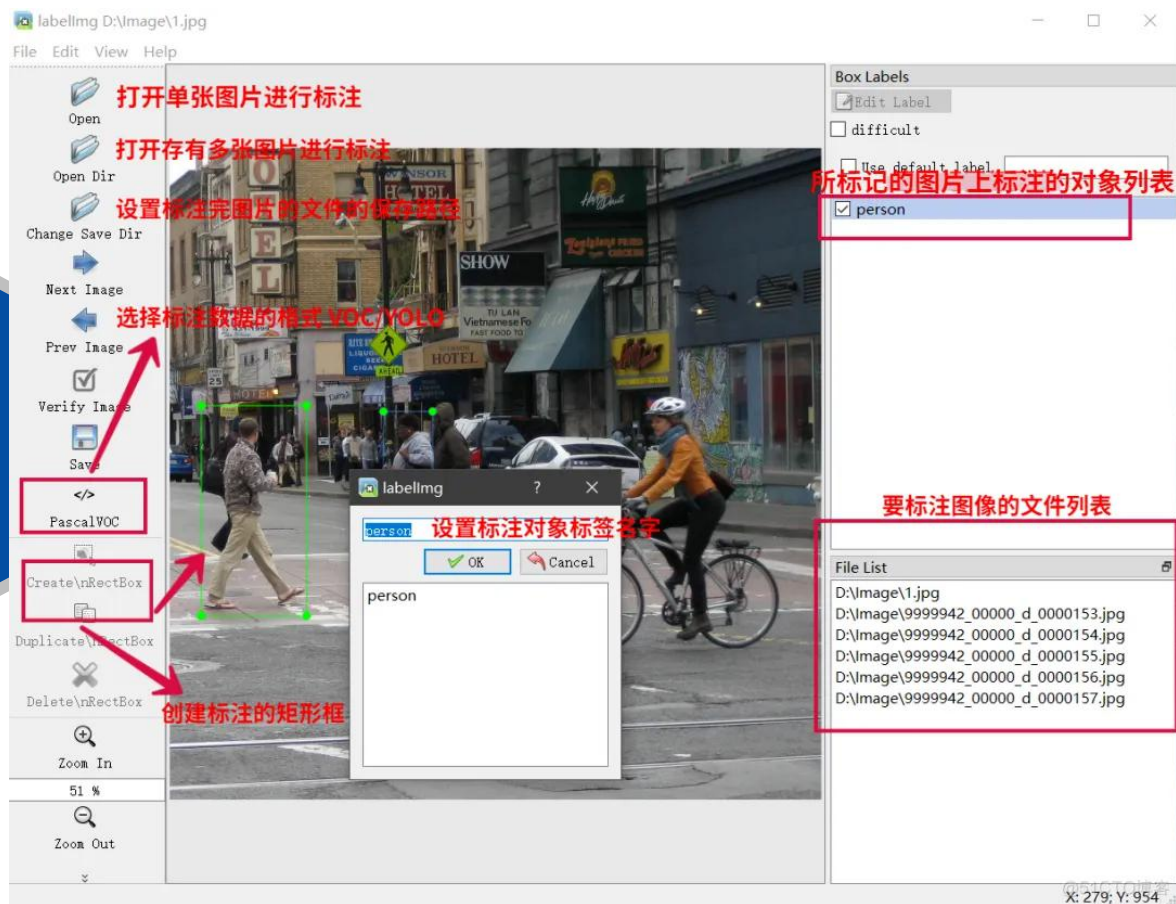
总体实现框架：

数据集制作——模型训练——部署到树莓派——界面设计——性能分析



PART THREE

技术路线



数据集制作

- 编写程序，使用树莓派4B的 Camera 直接在捕获需要识别目标物的图片信息。
- 将捕获到的待识别目标物照片传输到PC端，确定类别，利用 Labelme 软件进行标注。
- 编写程序，将标注的文件JSON格式转txt的Yolo格式。

存储图片的文件夹名称、图片名称、图像存储的完整路径

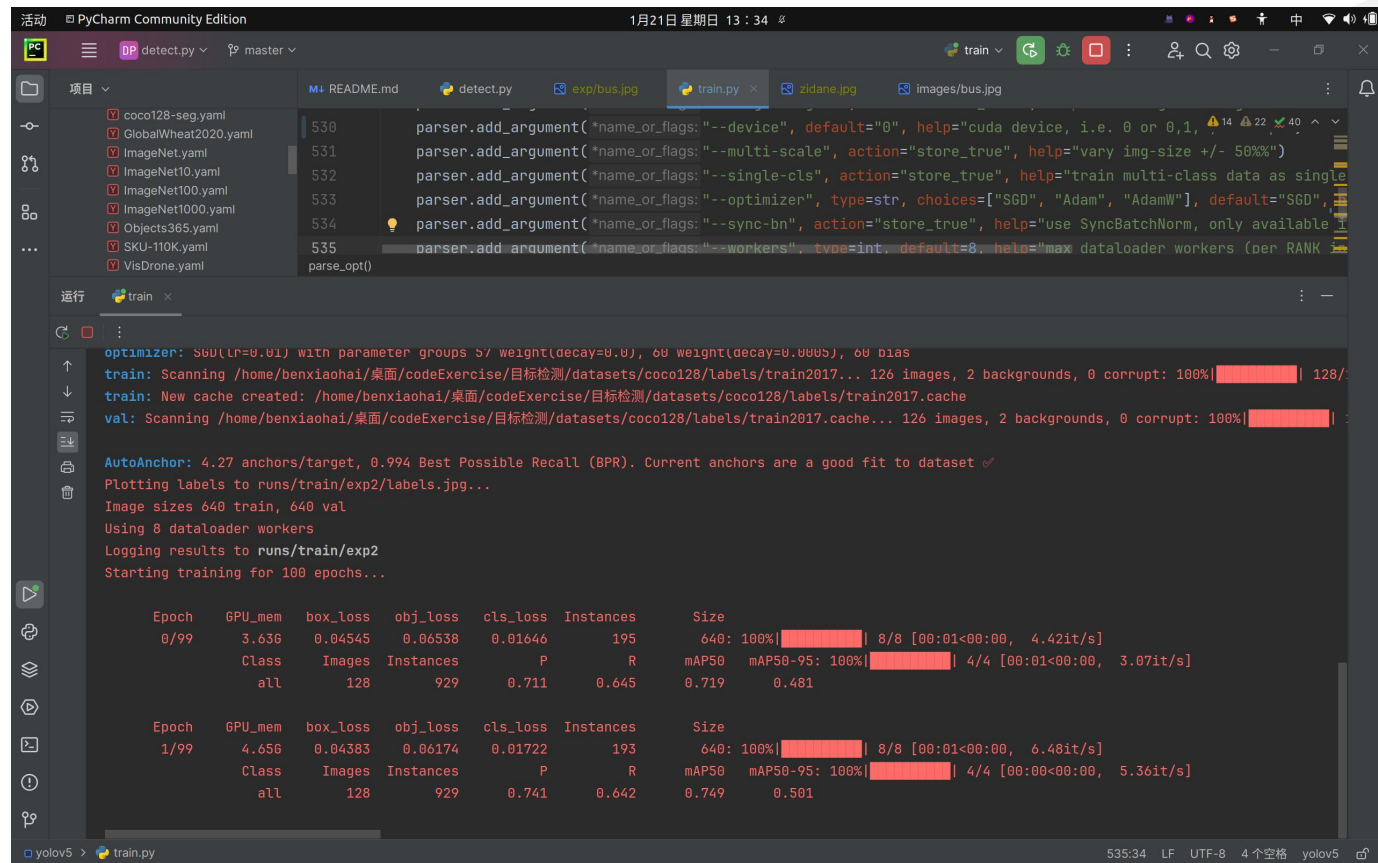
```
<?xml version='1.0'?>
<annotation>
  <folder>Image</folder>
  <filename>9999942_00000_d_0000152.jpg</filename>
  <path>D:\Image\9999942_00000_d_0000152.jpg</path>
```

图片上标注的目标的基本信息（目标名，框柱目标的框的左上角和右下角坐标位置）

```
  <source>
    <database>Unknown</database>
  </source>
  <size>
    <width>1330</width>
    <height>1330</height>
    <depth>3</depth>
  </size>
  <segmented>0</segmented>
  <object>
    <name>person</name>
    <pose>Unspecified</pose>
    <truncated>0</truncated>
    <difficult>0</difficult>
    <bndbox>
      <xmin>64</xmin>
      <ymin>527</ymin>
      <xmax>279</xmax>
      <ymax>964</ymax>
```

目标检测模型训练

- 从GitHub上下载Yolov5、Yolov5-Lite的源代码，在Pycharm中配置模型的环境。
- 将制作好的数据集放到训练对应的目录下。
- 配置训练集、测试集地址、种类数及相应的标签。
- 训练模型，得到训练数据内容 (best_weight、last_weight等)



```
python train.py
```

optimizer: SGD(lr=0.01) with parameter groups 5/ weight(decay=0.0), 68 weight(decay=0.0001), 68 bias

train: Scanning /home/benxiaohai/桌面/codeExercise/目标检测/datasets/coco128/labels/train2017... 126 images, 2 backgrounds, 0 corrupt: 100%|██████████| 128/

train: New cache created: /home/benxiaohai/桌面/codeExercise/目标检测/datasets/coco128/labels/train2017.cache

val: Scanning /home/benxiaohai/桌面/codeExercise/目标检测/datasets/coco128/labels/train2017.cache... 126 images, 2 backgrounds, 0 corrupt: 100%|██████████|

AutoAnchor: 4.27 anchors/target, 0.994 Best Possible Recall (BPR). Current anchors are a good fit to dataset ✓

Plotting labels to runs/train/exp2/labels.jpg...

Image sizes 640 train, 640 val

Using 8 dataloader workers

Logging results to runs/train/exp2

Starting training for 100 epochs...

Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
0/99	3.636	0.04545	0.06538	0.01646	195	640: 100% ██████████ 8/8 [00:01<00:00, 4.42it/s]
	Class	Images	Instances	P	R	mAP50 mAP50-95: 100% ██████████ 4/4 [00:01<00:00, 3.07it/s]
	all	128	929	0.711	0.645	0.719 0.481

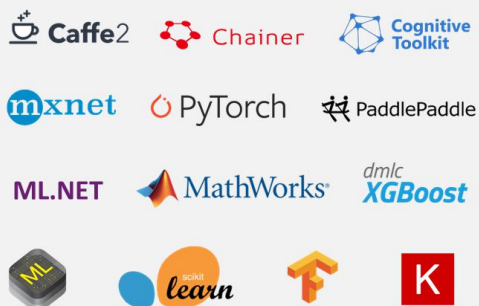
Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
1/99	4.656	0.04383	0.06174	0.01722	193	640: 100% ██████████ 8/8 [00:01<00:00, 6.48it/s]
	Class	Images	Instances	P	R	mAP50 mAP50-95: 100% ██████████ 4/4 [00:00<00:00, 5.36it/s]
	all	128	929	0.741	0.642	0.749 0.501

PART THREE

技术路线

Create

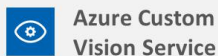
Frameworks



Native support

Converters

Services

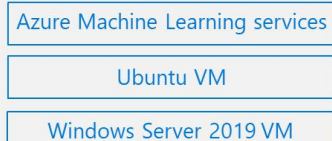


Native support



Deploy

Azure



Windows Devices

Other Devices (iOS, etc)

模型部署

- 将模型转换成ONNX模型。
 - ONNX: Open Neural Network Exchange, 支持许多框架，框架中的模型都可以被转换成ONNX格式，从而实现跨平台。
- 在树莓派上安装onnxruntime，用来成功运行ONNX格式的模型。



目标检测及性能展示

- 编写程序，调用摄像头，运行目标检测算法。
- 利用PyQT技术，为程序编写一个前端界面。分别展示摄像头的识别图像以及相应的性能指标 (FPS、准确率等)。



工作安排

前期进行：

- ① 对该选题相关的资料进行搜集整理，并学习深度学习基本理论、Yolov5、Yolov5-Lite、树莓派；
- ② 分析Yolov5、Yolov5-Lite的神经网络结构，尝试提出一种新的网络结构。

后期进行：

- ① 编写程序，完成数据集制作模块。
- ② 使用制作好的数据集训练模型。
- ③ 将模型部署到树莓派上，并为程序编写前端界面。
- ④ 论文写作

完成选题和《开题报告》

2024-01-03-2024-01-21

系统开发和论文写作

2024-02-13-2024-04-20

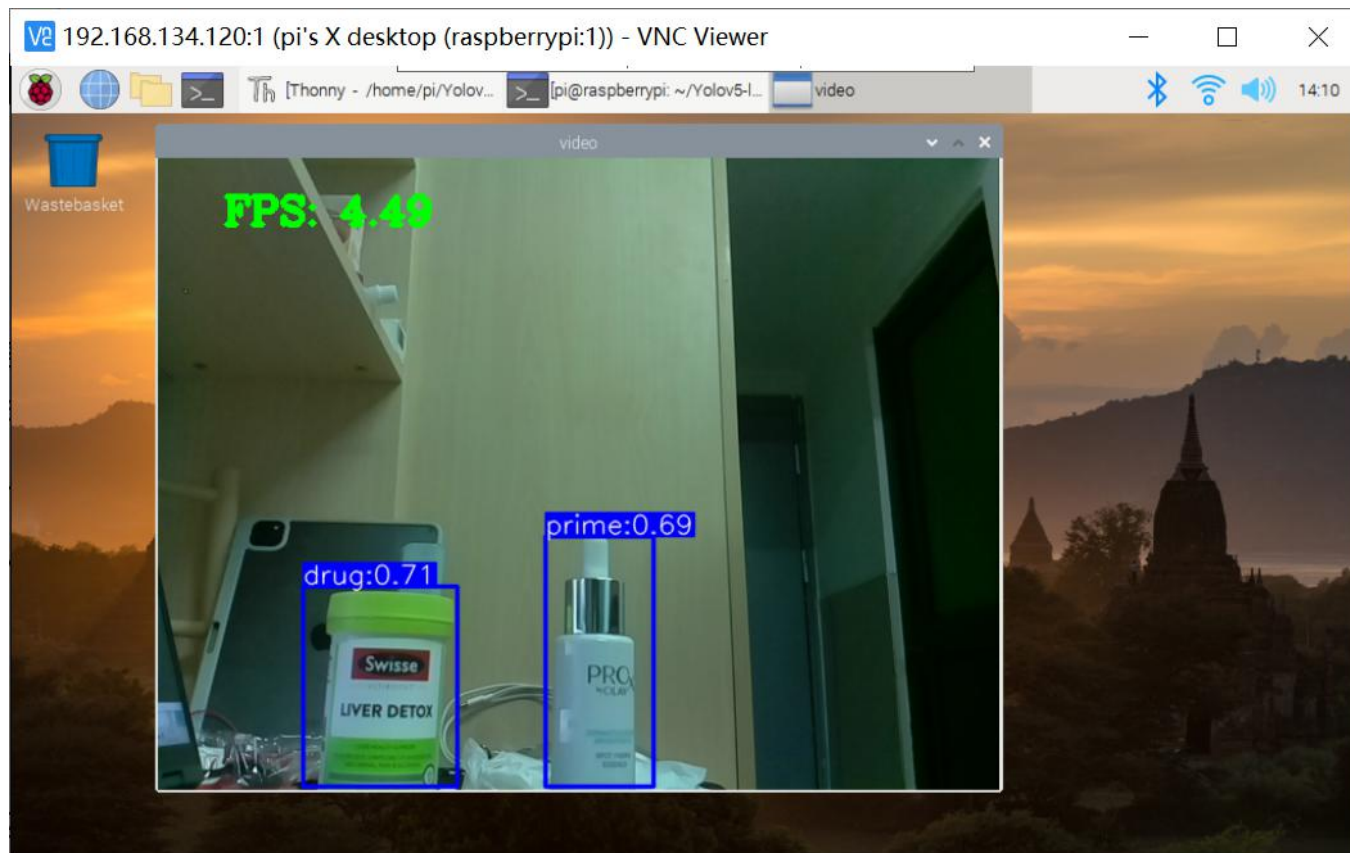
2024-1-22-2024-02-08

理论学习、完成数据集
制作模块、树莓派学习

2024-04-20-2024-05-31

论文查重和毕业答辩

预期效果



实现基于Yolov5、Yolov5-Lite目标检测。将训练好的模型部署到树莓派4B上，运行目标检测算法，分别显示其检测速度和准确率。

感谢观赏

THANKS FOR WATCHING

