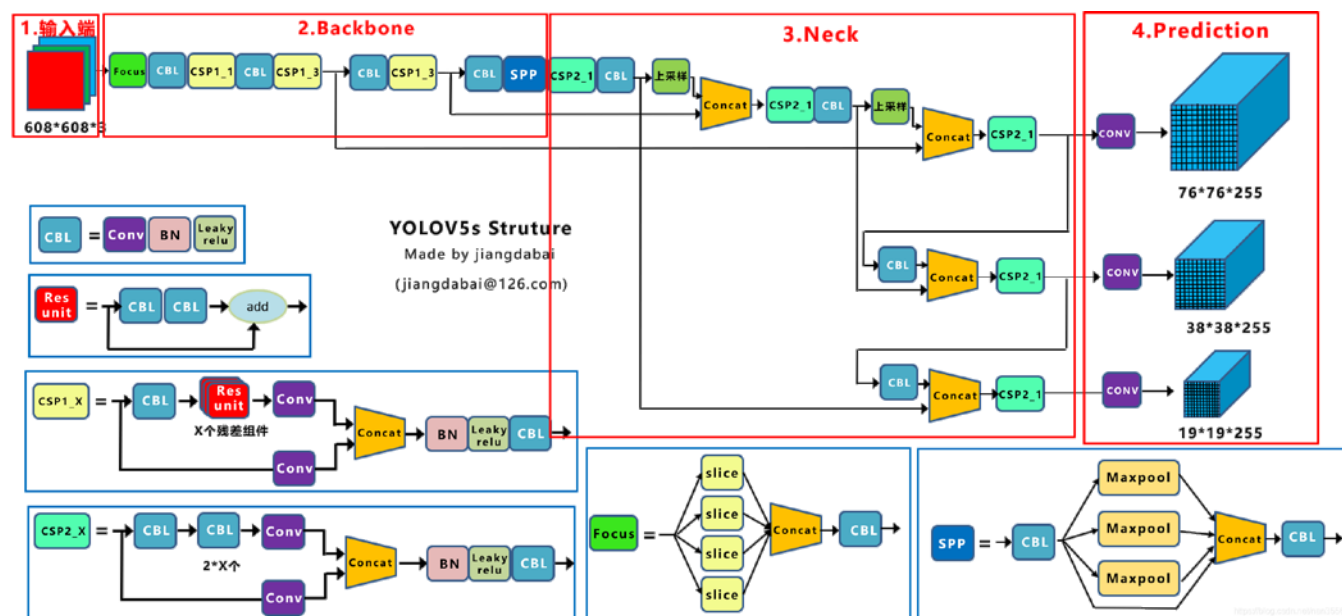
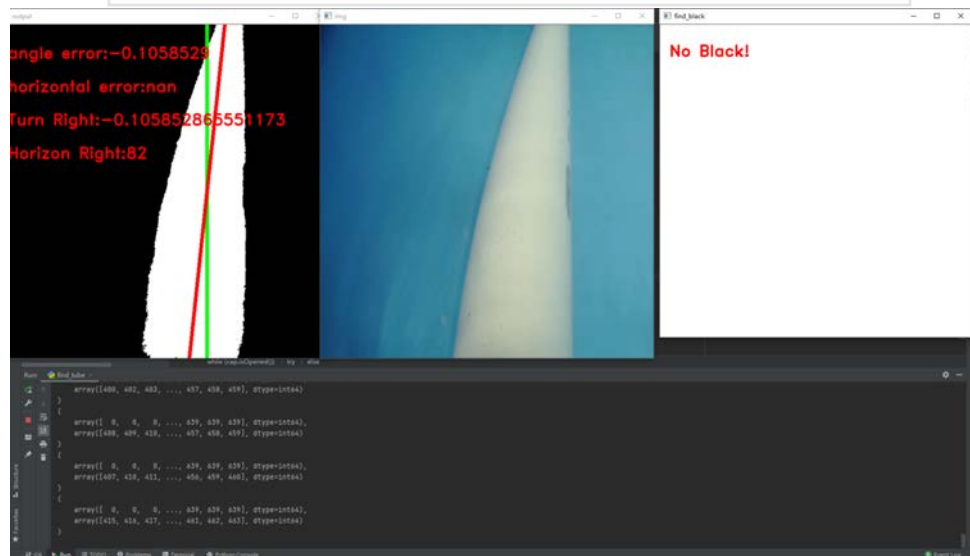
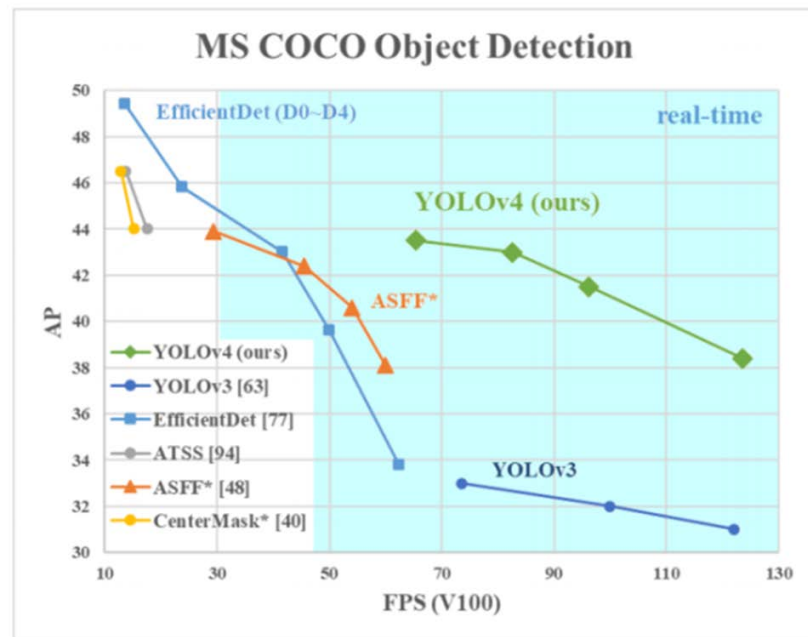


粤港澳大学生工程训练综合能力竞赛暨第七届全国大学生工程训练综合能力竞赛广东省分赛		<div>控制及电路设计方案</div> <div>Control and Electronic Circuit Design Scheme</div>	共 4 页	第 1 页
			编 号	(工作人员填写)
<div></div>		<div>1、检测及控制设计思路</div> <div>1) 检测： 检测管道：用 opencv 分别对 BGR 图片中的 G、R 颜色空间进行二值化，然后进行与运算得到比较稳定的二值化图像，从而可以检测出管道。得到管道图像后，对所有的白色点用 numpy 进行拟合可以得到管道简化为的走向图，然后用中心点创建垂直的直线判断管道是左倾还是右倾，然后用中心点与图像中间的间距判断机器人的水平位置偏移量。 检测黑色物块：用 yolov5 目标检测模型进行目标检测。 (2) 上位机控制： 上位机：Jetsonnano 信息获取：获取图像，并将图像传给检测代码并得到返回信息；从串口读取 stm32 传上来的深度信息和 IMU 方位角信息 过程控制：使用增量式 pid 对上述信息进行处理控制电机的转速，使用 ros 的串口通讯将电机转速传给 stm32 (3) 下位机控制： 下位机：STM32F103ZET6 最小系统板 信息获取与发送：通过设计通讯数据包，使用 STM32 与上位机 Jestonnano 进行串口通讯，获取六个推进器的转速和控制标志，将转速传给推进器实现运动；同时 STM32 分别通过串口中断和轮询获取 IMU 的欧拉角信息和深度传感器的深度信息，并将其通过串口发送给上位机： 过程控制：通过 STM32 定时器产生 PWM 波输出给推进器连接的电调，驱动推进器工作</div>		
<div></div>		<div>2、器件选择及实施方案</div> <div>1. 推进器：选用与 30A 电调相连的 2216 型推进器马达, 通过 STM32 输出 50HzPWM 波进行控制 2. IMU: 选用 HiPNUC 生产的 9 轴姿态传感器 HI229, 进行配置后利用串口中断读取其输出的欧拉角信息 3. 深度传感器：选用 MS5837 压力传感器，通过串口轮询读取其输出的温度和深度信息 4. 下位机：选用 STM32F103ZET6 最小系统板 5. 电源：选用 11.6V 的 4S 电池为推进器供电，使用 5V 输出的充电宝为 Jestonnano 供电，STM32F103 的供电由 Nano 引出，其他传感器的电源从 STM32 引出</div>		



3、总结和体会

(1) 管道检测：首先采用的想法是用图像分割模型，例如 Unet，DeeplabV3+等，但是运行后发现对算力要求太大，如果同时使用目标检测模型和图像分割模型，一秒仅有几帧图像，远远无法满足检测要求。而且用 opencv 进行传统的形态学处理后发现得到的结果几乎和图像分割没有太大区别。所以我们放弃了图像分割模型，使用形态学处理+目标检测 yolov5 模型。形态学处理采用了分离颜色空间，再进行二值化+与运算而非直接灰度处理，高斯模糊+二值化。前者得到的图像更加稳定。然后用 numpy 的 polyfit 进行拟合求解方向。

(2) 物体检测

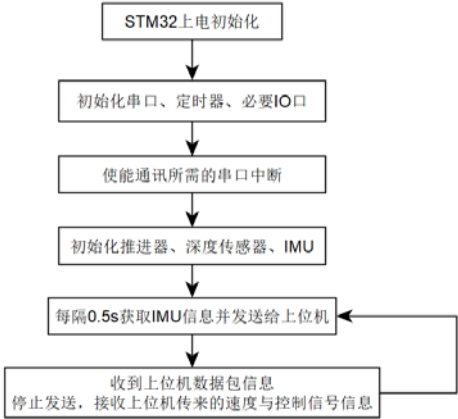
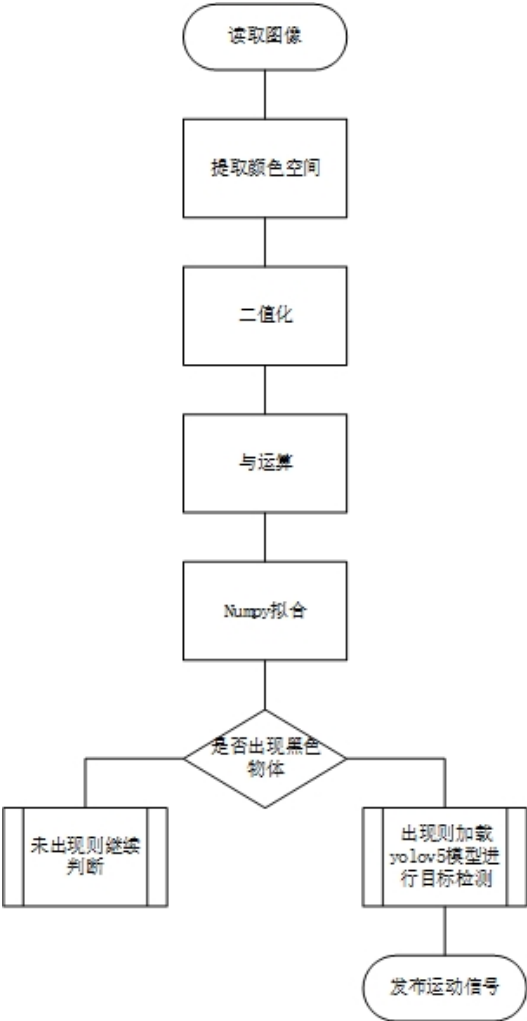
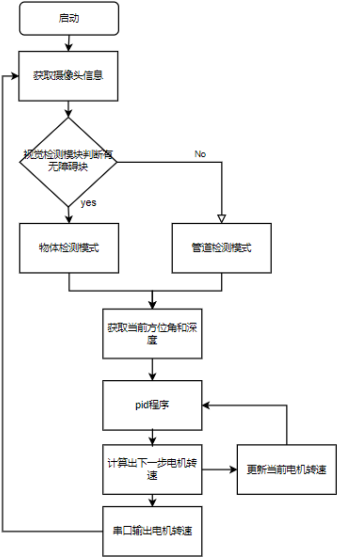
物体检测采用了新发布的 yolov5 目标检测模型。首先，这是 YOLO 家族中，第一个先用 PyTorch 的原生版本，而非 PJ Reddie 的 Darknet 编写的模型。Darknet 是一个非常灵活的研究框架，但它并没有考虑到生产环境的构建，而且用户社区也较小。这导致 Darknet 需要在配置上花费不少功夫，而且生产准备不足。

由于 YOLOv5 是在 PyTorch 中实现的，它受益于成熟的 PyTorch 生态系统：支持更简单，部署更容易。此外，作为一个更广为人知的研究框架，YOLOv5 的迭代对更广泛的研究社区来说可能更容易。这也使得部署到移动设备上更加简单，因为该模型可以轻松编译成 ONNX 和 CoreML。

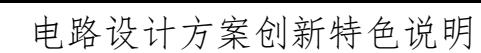
其次，YOLOv5 的速度快得惊人。在 YOLOv5 Colab notebook 上，运行 Tesla P100，每张图像的推理时间仅需 0.007 秒，这意味着每秒 140 帧（FPS）。相比之下，YOLOv4 在被转换到相同的 Ultralytics PyTorch 库后的速度是 50FPS。YOLOv5 的速度是 YOLOv4 的 2 倍还多。

第三，YOLOv5 精度超高。在 Roboflow 对血细胞计数和检测（BCCD）数据集的测试中，只训练了 100 个 epochs 就达到了大约 0.895 的平均精度（mAP）。诚然 EfficientDet 和 YOLOv4 的性能相当，但在准确率没有任何损失的情况下，看到如此全面的性能提升是非常罕见的。

第四，YOLOv5 的体积很小。具体来说，YOLOv5 的权重文件是 27 兆字节。YOLOv4（采用 Darknet 架构）的权重文件是 244 兆。YOLOv5 比 YOLOv4 小了近 90%。这意味着 YOLOv5 可以更容易地部署到嵌入式设备上。

<div>STM32 控制流程图：</div>  <div>视觉控制流程图：</div>  <div>Jetson nano 控制流程图</div> 		
<div>说 明</div> <p>（1）STM32 控制部分：</p> <p>STM32 主要负责推进器的驱动和传感器信息的获取，在程序中，首先对 STM32 各项基本外设进行初始化，然后通过 IO 口和串口对推进器和传感器进行初始化配置，配置完成后，STM32 定时获得传感器数据，并通过串口通讯协议以数据包的形式发送给上位机。同时，STM32 会持续等待上位机发送的速度和状态控制信息，并更新对推进器的控制信号</p> <p>（2）视觉控制部分：</p> <p>由摄像头不断捕获水下图像，然后用 OpenCv 提取 G、R 颜色空间并进行二值化，然后进行与运算，得到管道为白，其余为黑的单通道二值化图像。随后进行 numpy 的直线拟合，得到管道的走势和水平偏移，从而控制机器人运动。在运动过程中对黑色点进行判断，达到一定阈值即判断为有黑色物体，然后加载 yolov5 模型文件进行目标检测，进行蜂鸣器报警和 RGB 灯闪烁。</p> <p>（3）jetson nano 控制部分：</p> <p>1.信息获取：摄像头信息，通过串口通讯从 stm32 获取的方位角信息和深度信息</p> <p>2.管道检测模式：将信息传给视觉检测程序获取当前管道位置，将管道位置和从 stm32 获取的信息和当前电机转速传入 pid 控制程序中，获取下一步的电机转速。</p> <p>3.物体检测模式：当视觉检测程序检测到图像中有物块时，程序会实时检测目标物块位置，并在机器人到达物块正上方时停下来，进行报警和识别，并控制机器人下潜完成回收操作。</p>		
电路程序流程图及说明	比 例	1： 1
粤港澳大学生工程训练综合能力竞赛暨第七届全国大学生工程训练综合能力竞赛广东省分赛	共 4 页	第 3 页

- 3 -



理由如下：

- (1) 引出 TIM2-TIM3 的 PWM 通道且安排与相近位置, 方便推进器驱动信号统一接线
- (2) 引出五路串口 UART1-UART5, 且大部分串口 RX-TX 引脚位置相邻, 方便与通讯模块连接
- (3) 核心板预留出 5V 与 3.3V 电源输出, 便于给深度传感器和 IMU 供电, 无需再引入电源模块

电路图及特色说明	比 例	1: 1
粤港澳大学生工程训练综合能力竞赛暨第七届全国大学生工程训练综合能力竞赛广东省分赛	共 4 页	第 4 页