

华南虎视觉组提前批任务

1. 装甲板目标精确识别任务

在RoboMaster系列赛事中，对战的机器人通过发射弹丸攻击对方。而机器人具有感知弹丸击打的装甲模块，在图像上是由两个发光灯条和数字构成的。视觉组最最基本的作用，就是使用光学传感设备以及运算平台，对装甲板的特征进行识别、筛选、决策，完成瞄准的功能——自瞄。这是RM视觉里最传统，但最重要的功能。



【任务】使用指定测试视频，对上述提及到的装甲板进行识别、决策。

要求

1. **必须**使用C++作为编程语言，**不得**使用Python，视觉库使用：OpenCV
2. 技术要求：**必须**使用C++11及以上版本的C++特性，代码结构上**必须**使用类对灯条、装甲板进行封装，在不导致过度耦合的情况下，可适当使用继承、多态性。
3. 程序能够正常运行，且具备较强的抗干扰能力

加分项（三选一）

1. 实际上，光靠识别是无法控制机器人完成自瞄动作的，因此，需要借助串口通信来完成视觉-电控的通信，从而达到视觉控制机器人云台pitch、yaw角度的功能。请实现Ubuntu环境下的串口通信，自定义一个串口通信协议包，并且满足接收端（可以是另外一台笔记本电脑）能够正常接收自己发送过去的的数据协议包。
2. 由于实际过程中，装甲板不是一直静止的，此时发弹肯定会有所滞后，并无法击中装甲板目标，请设计一款预测器，能够对装甲板的运动轨迹进行预测（在短时间内当成匀速运动即可）。并且在图像中画出预测出来的装甲板中心点。
3. 由于实际过程中，瞄准装甲板后子弹受重力影响会有一定的下坠，且跟距离有关。请对装甲板距离进行计算，设计一款补偿器，在给定的条件下计算出因重力下坠所需要的抬枪补偿（射速15m/s，装甲板与发射器在同一水平线上）。

2. CMake框架搭建任务

CMake是一个跨平台的安装（编译）工具，可以用简单的语句来描述所有平台的安装(编译过程)。一般在大型项目开发中，CMake是必不可少的。华南虎视觉组组长在长期的技术积累后已经搭成了一套相对完善的华工机器人实验室视觉库（SRVL）。为此，想读懂此视觉库，并将其作为工具编写有关各个机器人的视觉代码，就需要熟练的使用CMake。

【任务】给定一个代码框架：[Git代码仓库地址](#)，尝试创建缺少的 CMakeLists.txt 并编写，将此项目链接起来，编写过程中禁止改动任何*.h，*.hpp，*.cpp文件，仅允许修改或创建 CMakeLists.txt。最终能够按照给定的方式运行可执行文件。

要求

【CMake 部分变量命名要求】

- CMake 最小版本号：3.10
- 项目名：Test，可执行文件名：test
- 库目标名：与文件夹名一致 例如：

```
1  add_library(ABC ${ABC_DIR})
2  # 其中 ABC 文件夹如下
3  # ABC
4  # |— include
5  # |   |— ABC.h
6  # |— src
7  #     |— ABC.cpp
```

- 部分 CMakeLists.txt 已给出，禁止修改，包括：
 - CMake_Test/common/kalman/CMakeLists.txt
 - CMake_Test/common/CMakeLists.txt
 - 部分CMake_Test/CMakeLists.txt
- 需要提前安装好OpenCV

【编译运行方式】

```
1  mkdir build
2  cd build
3  cmake ..
4  make -j6
5  ./test
```

3. ROS联合仿真任务

[ROS](#)是一组软件库和工具，可帮助您构建机器人应用程序。从驱动程序到最先进的算法，以及强大的开发人员工具，ROS可以满足您的下一个机器人项目的需求。而且这都是开源的。



Get ROS Noetic
Ninjemys on Ubuntu
Linux

(Recommended for
Latest ROS 1 LTS)

[Install](#)



Get ROS Foxy Fitzroy on
Ubuntu Linux, macOS, or
Windows 10

(Recommended for Latest
ROS 2 LTS)

[Install](#)



Get ROS Galactic
Geochelone on Ubuntu
Linux, macOS, or Windows
10

(Recommended for Latest
ROS 2)

[Install](#)

【任务】在你的Ubuntu系统上安装对应的ROS版本，并学习其运行框架和通信机制。基于以上的装甲板识别任务后，利用ROS的分布式节点通信机制，实现在RVIZ上的三维仿真效果。

要求

1. 独立思考，阐述解决问题的完整过程
2. 实现流畅、完整的仿真过程

4. 深度学习任务

- [PyTorch](#)是一个Python 优先的深度学习框架，可以实现基于GPU 加速的Tensor计算和构建深度神经网络等。



- [MNIST数据集](#)是美国国家标准与技术研究院收集整理的大型手写数字数据库,包含60,000个示例的训练集以及10,000个示例的测试集。对于刚入门深度学习的同学来说，通常是最先接触到的数据集，相当于编程中的Hello World！

【任务】尝试用PyTorch自行构建神经网络，将其用于MNIST的训练与测试中，并达到识别率99.7%以上的测试效果

要求

1. 阐述网络结构的设计思路，以及在优化识别率中做的每一步改进策略
2. 熟悉使用PyTorch训练权重的整个流程
3. （有余力者做）尝试将训练出的权重其部署到C++，实现对手写数字图片的预测

完成要求

- 所有任务应当在Ubuntu 20.04环境下完成
- OpenCV版本保持在4.4.0以上
- 提前批队员之间可相互探讨，但不可有代码间的直接交流接触（经发现取消评选资格）

提交形式

- 注册一个github或gitee账户，并创建一个名为【23-视觉-姓名】的仓库，将所有提交材料上传到git仓库，最终提交仓库地址即可
- 每个任务都需要有充分的展示效果，尽可能以录屏或截图的形式，任务1、3必须有完整的录屏
- 提交一份技术报告文档（有模板），详细阐述解决问题的完整过程，尽可能体现自己思考和分析问题的能力