

智能车实验室 2025 年招新竞速组视觉作业

作业一：OpenCV 在 ROS 中的使用

任务目标：使用 C++构建两个完整的 ROS 包（通过 launch 文件启动），通过 cv_bridge 实现以下功能：

- 1.图像发布节点：将本地存储的图像 input.jpg 加载为 OpenCV 图像 (cv::Mat)，并将其转换为 ROS 图像消息 (sensor_msgs/Image)，然后发布给其他 ROS 节点。
- 2.图像订阅节点：订阅接收到的 ROS 图像消息，将其转换为 OpenCV 图像 (cv::Mat) 格式，处理后保存为本地文件 output.jpg。

命名要求：

要求两个 ROS 功能包符合完整的格式（输入输出的两张图像分别放置在各自功能包文件夹的 config 文件夹下，config 与 src 文件夹同级）；

图像发布功能包命名为 image_publisher, 图像接受功能包命名为 image_subscriber; 节点命名为 image_publisher 和 image_subscriber，话题发布到/publisher 和 /subscriber。

提交格式：将所有源码文件夹统一放在 test1 文件夹中，将执行过程中的完整命令行截图保存为 test1/命令行.jpg 或者.png。

作业二：基于压缩节点的图像编码传输

任务目标：使用 C++构建两个完整的 ROS 包（通过 launch 文件启动），通过 ROS 提供的的压缩图像话题 (sensor_msgs/CompressedImage) 实现以下功能：

实时视频发布节点：从本地摄像头或视频文件(可以自己准备一段长一点的视频)中实时读取图像帧，将其转换为 ROS 压缩图像消息，并发布到话题 /compress_publisher。

命名要求：在实现实时视频发布过程中，所有的参数必须通过 config/param.yaml 配置文件通过 launch 文件传递，源码中禁止硬编码调参。参数配置需包括以下内容：①摄像头画面 640*480 的限制；②帧率为 30 帧左右；③对比度、亮度调节（随机调节即可）、压缩图像压缩为 30%；

提交格式：将所有源码文件夹统一放在 test2-3 文件夹中，将执行过程中的完整命令行截图保存为 test2-3/命令行.jpg 或者.png。

作业三：无图形化界面的视频流展示

任务目标：使用 python 或者 C++编写一个图像接收功能包，模拟无 GUI 界面并将这个视频流发布在本地和局域网某端口上，[接收的是作业 2 中压缩图像](#)。

命名要求：端口号使用[学号后四位](#)对应的端口；代码中禁止出现 GUI 相关的函数（如 OpenCV 的 imshow 等）；不允许[直接套用某个库](#)（如 flask），请自行优化（例如优化传播速度、画面、网络通信等）。

提交格式：将所有源码文件夹统一放在 test2-3 文件夹中，将执行过程中的[完整命令行截图](#)保存为 test2-3/命令行.jpg 或者.png；同时将执行过程中的[浏览器查看端口的截图](#)保存为 test2-3/端口.jpg 或者.png。

作业四：windows 上的 Yolo v5 模型的使用尝试

任务目标：

目标一：使用数据集中的所有图像进行模型训练，将数据集按照 8:1:1 的比例随机分为 train、valid、test（[test 不放入模型中训练](#)）

目标二：训练 train、valid 得到一个 yolov5 的 best 和 last 的 pth 模型，[将 best.pth 转化为 best.onnx 模型](#)（放在和 pth 文件的同级目录下），

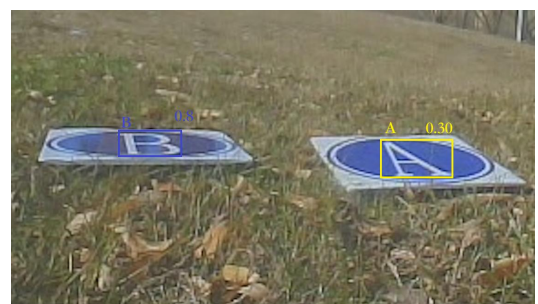
目标三：写一个 python 脚本 test.py，使用 best.onnx 处理 test 中的所有图像，自己[编写后处理](#)，并将[输出的带有识别框的图像](#)保存在 test_result 文件夹里。

命名要求：

要求一：标签类别为 A 和 B，

要求二：test_result 中输出图像的命名为“源文件名_test.jpg”；

例如：对 00001.jpg 进行处理，将识别框绘制在图像上，并在框的左上角显示类别名，右上角为置信度；A 使用黄色，B 使用蓝色标记。



提交格式：将本题中的以下文件保存在 test4 文件夹中：训练中输出的 run 文件夹、test.py、test_result 文件夹。

把上面的 test1、test2-3、test4 文件夹统一放置在以[学号](#)命名的文件夹下并压缩。