Depth Estimation Deployment (Python)

1. 模型转换：torch -> onnx -> tensorrt engine

1. 加载训练好的模型
2. torch -> onnx -> tensorrt serialized engine

2. 获取设备输入（RealSense D435i）

1. 安装 realsense sdk 后可以直接用命令 realsense-viewer 查看输出
    1. 参考链接:https://github.com/IntelRealSense/librealsense/blob/master/doc/installation_jetson.md
2. python 接口依赖包：pyrealsense2
3. 建立捕获视频流的 pipeline，config 设置（画面宽/高/帧率等）
    1. 注意配置的 depth_camera 和 rgb_camera 分辨率不是随意的，可能导致错误 （Couldn't resolve requests - "self.pipeline.start(config)"），可以打开 realsense-viewer 查看支持的分辨率设置
4. 捕获 RGB 和 Depth 视频流，并转为 numpy array (uint8)

3. 预处理

1. numpy dtype 转换：uint8 -> fp32
2. 归一化 （0-255 -> 0-1）
3. resize
4. 归一化（指定均值和方差）
5. 调整通道排列 （HWC -> CHW）
6. 保证内存连续性
7. 扩展维度满足模型输入要求

4. 模型推理（on jetson orin nx 16GB）

1. 加载 tensorrt engine
2. 在 host：CPU， device: GPU 上为输入输出分配内存
3. 创建 cuda stream
4. 将预处理后的图像数据展平并复制到固定内存
5. 将数据从固定内存异步复制到GPU内存
6. 在GPU上异步执行推理操作
7. 将推理结果从GPU内存异步复制回固定内存

8. 同步CUDA流以确保所有异步操作完成

5. 后处理 & 可视化

1. numpy reshape: (132496,) -> (364, 364)
2. cv2 resize: (364, 364) -> (480, 640)
3. 归一化到 0-255
4. numpy dtype 转变：fp32 -> uint8
5. cv2 深度图映射为彩色图 (480, 640) -> (480, 640, 3)
6. 拼接rgb/depth_pred/depth_gt 可视化输出