Objective:
Develop a pallet & gound detection-segmentation application in ROS2 for a manufacturing or warehousing environment. The solution should be optimized for deployment on edge devices like the NVIDIA Jetson AGX Orin, ensuring real-time performance suitable for mobile robotics applications.

Tasks:

1. **Dataset Acquisition and Preparation:**
    - **Dataset Recommendation:** Use the link to access open source database for pallets in different scenarios { Pallets }
    - **Data Preparation:**
        - Annotate **pallets** & **ground** using existing annotation tools (ex. DINO)
        - Organize the dataset into training, validation, and test sets.
        - Apply data augmentation techniques (e.g., varying lighting conditions) to simulate real-world scenarios.

2. **Object Detection and Semantic Segmentation:**
    - **Model Development:**
        - Implement an object detection model (e.g., **YOLOv11** etc.) to identify pallets.
        - Develop a semantic segmentation model to segment pallets and ground
        - Train and fine-tune the models using your prepared dataset.
    - **Performance Evaluation:**
        - Assess the models using metrics like mAP for detection and IoU (Intersection over Union) for segmentation.
        - Ensure models perform robustly under varying environmental conditions.

3. **ROS2 Node Development:**
    - **ROS2 Package:**
        - Develop a ROS2 package with nodes written in **Python**, **C++**, or **RUST**.
        - Nodes should:
            - Subscribe to image and depth topics from a simulated or real camera.
            - Perform object detection & segmentation
        - To streamline the assessment please make sure you include a README, ros2 inference node works off the shelf with the

following camera data and detections/segmentation plotted on the original image.
    - Use humble as your ros2 distribution.

4. **Edge Deployment Optimization (Optional) :**
    - **Model Optimization:**
        - Convert your models to formats suitable for edge deployment (e.g., **TensorRT**, **ONNX**).
        - Apply optimization techniques like quantization and pruning to enhance performance.
    - **Docker Container:**
        - Dockerized the complete module that can natively run on different devices as long as Nvidia drivers are present.

5. **Evaluation Criteria:**
    - We will run the module on AGX Orin and feed live camera data from zed 2i
    - Performance will be measured based on pallet detection accuracy under varying conditions

Create complete assignment as a GitHub repository and share with the team