

Deep Learning Pipeline + ROS 2 Deployment

This repository provides end-to-end code and infrastructure for pallet detection and ground segmentation using a combination of Python-based notebooks (for training and data preparation) and ROS 2 (for real-time inference deployment via Docker).

Repository Structure

```
├── NOTEBOOKS/
|   ├── PALLET_DETECTION_YOLOv8.IPYNB
|   ├── TWO_PALLET_DETECTION_YOLOv8_WITH_AUGMENTATION.IPYNB
|   ├── GROUND_SEGMENTATION_EFFICIENTNETB3_UNET.IPYNB
|   ├── TF2ONNX.IPYNB
|   ├── GROUND_SEGMENTATION_MASKS_PIPELINE - SAM.IPYNB
|   ├── DATASET_CLEANING.IPYNB
|   └── EUROPALLET - TEST MULTIPLE.IPYNB
|
├── ROS2_WS/
|   ├── SRC/PALLET_INFER_NODE/
|   |   ├── MODELS/          # BEST.ONNX, GROUND_SEG.ONNX
|   |   ├── STATIC_IMAGE_INPUT_FOLDER/
|   |   ├── OVERLAY_OUTPUT/
|   |   ├── PALLET_INFER_NODE.PY  # INFERENCE NODE
|   |   ├── STATIC_IMAGE_PUBLISHER.PY
|   |   ├── INFER_LAUNCH.PY    # LAUNCH FILE
|   |   └── ...
|   └── DOCKERFILE
|
```

├── REQUIREMENTS.TXT

└── README.MD

Notebooks Overview

1. Pallet Detection

- PALLET_DETECTION_YOLOv8.ipynb: YOLOv8 model training for pallet detection.
- TWO_PALLET_DETECTION_YOLOv8_with_AUGMENTATION.ipynb: Augmented training for multiple pallets.

2. Ground Segmentation

- GROUND_SEGMENTATION_EFFICIENTNETB3_UNET.ipynb: Ground segmentation using U-Net with EfficientNetB3 encoder.
- TF2ONNX.ipynb: Converts the trained .h5 U-Net model to .onnx format.

3. Segmentation Mask Generation

- GROUND_SEGMENTATION_MASKS_PIPELINE - SAM.ipynb: Uses SAM for manual ground mask creation.

4. Testing & Utilities

- EUROPALLET - TEST MULTIPLE.ipynb: Loads multiple test samples to evaluate model performance.
- DATASET CLEANING.ipynb: Removes duplicates, filters based on quality, and renames files.

ROS 2 + Docker Inference Pipeline

A full ROS 2 (Humble) based deployment setup is provided, supporting ONNX model inference and visualization.

Setup

- Place best.onnx and ground_seg.onnx in the models/ folder.
- Run inside a Docker container to avoid dependency conflicts.

Run Inference Inside Docker

```
# Load image on new machine
docker load -i ros2_workspace_image.tar

# Run container
docker run -it --rm \
  --network host \
  -v /tmp/.X11-unix:/tmp/.X11-unix \
  -e DISPLAY=$DISPLAY \
  ros2-humble-workspace

# Inside container:
cd /root/ros2_ws
source install/setup.bash
ros2 launch pallet_infer_node infer_launch.py
```

Final Note

This repository represents a robust and scalable deep learning + robotics pipeline that brings computer vision models into real-world robotic environments using ROS 2 and Docker. It is ideal for warehouse robotics, pallet handling, autonomous mobile robots (AMRs), and research prototypes.