**Rust Detection Model (Summary)**

**Objective**  
Build and train an end to end object detection model to detect rust in images using bounding boxes, going through phases such as Dataset Preparation, Model Fine-Tuning, Validation & Testing.

**Dataset & Labeling**

* 100 images provided, annotated in CVAT exported as (YOLO Ultralytics Detection 1.0).
* Single class: rust.
* Split: 70/20/10 (train/val/test).

**Environment**

* Google Colab, GPU: NVIDIA Tesla T4.
* Framework: PyTorch via Ultralytics YOLO.

**Model & Training**

* Model: Tested **YOLOv8 (s/m)** and **YOLO11 (n/s/m)**, best results with **YOLO11m**.
* **Input size: 832** (to better capture small rust)
* **Epochs**: Trained up to 150 epochs with early stopping, triggered at (128).
* **Augmentations:** Applied color jitter (HSV), geometric transforms (small rotations, translations, scaling, shear, perspective), horizontal flips, low mosaic, disabled mixup, and random erasing to simulate occlusion while preserving rust textures.

**Challenges & Solutions**

* **Small dataset (100 images):** this small size can lead to overfit but this mitigated with regularization, augmentation, and early stopping.
* **Model selection:** tried multiple YOLO versions; YOLO11m gave the best balance of accuracy and efficiency.
* **Labeling:** labeling was repeated more than once to improve results. The first iteration was inconsistent due to rust nature and irregular shapes. This was corrected by adopting a clearer policy such being more precise, combining nearby rust areas into a single box when appropriate, and clearly defining what counts as rust to ensure consistency, this significantly improved model performance.
* **Low-contrast rust: difficult to detect at IoU 0.5; higher input size and tuned augmentations helped, but it remains a challenge**.

**Results and conclusion**

* The checkpoint was (epoch 78) reached **P=0.52, R=0.44, mAP@50=0.42** on validation.
* The model learned to detect rust but was limited by the small dataset.
* It shows no strong overfitting or underfitting, performance is mainly **data-limited**, and could be improved by increasing the number of images, ensuring diversity, consistent labeling, or segmentation annotation which is better to detect rust due to its spreading nature and its irregular shapes.