# 1. Objective

The objective of this project was to perform object detection using the YOLOv8 (You Only Look Once version 8) model on a custom dataset named "robotiq\_yolov11". The goal was to train the model to detect specific objects of interest and evaluate its performance on a validation set.

# 2. Dataset Description

Path: /content/dataset/robotiq\_yolov11/

Structure:

robotiq\_yolov11/  
├── data.yaml  
├── images/  
│ ├── train/  
│ └── val/  
├── labels/  
│ ├── train/  
│ └── val/

Classes: Defined in data.yaml with the appropriate number of object categories and names.

# 3. Environment Setup

Platform: Google Colab

Libraries Used:  
- ultralytics for YOLOv8  
- zipfile for extracting the dataset  
- matplotlib and built-in show() for displaying results

# 4. Training Configuration

Model: YOLOv8n (nano version for fast training/testing)  
Epochs: 50  
Image size: 640x640  
Batch size: 16

Training command:

from ultralytics import YOLO  
model = YOLO('yolov8n.pt')  
model.train(data='/content/model2/robotiq\_yolov11/data.yaml', epochs=50, imgsz=640, batch=16)

# 5. Inference

Inference was performed on the valid/images directory. The .show() method was used to visualize predictions for each image individually.

Inference command:

results = model('/content/dataset/robotiq\_yolov11/valid/images')  
for r in results:  
 r.show()

# 6. Performance Metrics

The model was evaluated using model.val() after training. Standard YOLO metrics such as mAP (mean Average Precision), precision, recall, and loss were generated.

# 7. Conclusion

The YOLOv8 model was successfully trained on a custom dataset for object detection. The predictions were visualized and verified to be accurate on the validation set. YOLOv8 provided a simple and efficient pipeline for training and inference on custom object detection tasks.

# 8. Future Work

Future improvements could include:  
- Using YOLOv8m or YOLOv8l for higher accuracy  
- Exporting model predictions to video or JSON  
- Deploying the model using a web interface for real-time applications