Underwater Object Detection Using YOLOv8

1. Overview

This project aims to detect underwater objects using the YOLOv8 deep learning model. It focuses on identifying marine animals in challenging underwater conditions such as poor visibility and noisy images.

---

2. Objectives

- Build an object detection system using YOLOv8.
- Train the model on a custom underwater dataset with 7 object categories.
- Compare YOLOv8 with other models like YOLOv5, YOLO-NAS, and RT-DETR.
- Evaluate results using precision, recall, F1-score, and mAP50.

---

3. Tools and Technologies

- **Language:** Python
- **Frameworks:** PyTorch, OpenCV
- **Models Used:** YOLOv8, YOLOv5, YOLO-NAS, RT-DETR
- **Annotation Tools:** LabelImg / Roboflow
- **System:** Intel Core i7 (12th Gen) CPU

---

4. Dataset

The dataset contains labeled images of the following underwater objects:

- Fish
- Jellyfish
- Shark
- Stingray
- Penguin
- Puffin
- Starfish

Annotations are in YOLO format.

5. Folder Structure

bash

CopyEdit

project/

├── dataset/ # Images and labels

├── models/ # YOLOv8 and other model weights

├── src/

│ ├── train.py

│ ├── evaluate.py

│ └── inference.py

├── sample_images/ # Test images

├── requirements.txt

└── README.md

---

6. Installation

bash

CopyEdit

git clone https://github.com/your-username/underwater-object-detection-yolov8.git

cd underwater-object-detection-yolov8

pip install -r requirements.txt

---

7. Training

bash

CopyEdit

python src/train.py --data dataset/data.yaml --epochs 100 --model yolov8n.yaml

---

8. Evaluation

bash

CopyEdit

python src/evaluate.py --weights models/yolov8.pt --data dataset/data.yaml

---

9. Inference

bash

CopyEdit

python src/inference.py --weights models/yolov8.pt --source sample_images/test1.jpg

---

10. Results

Model	mAP50	F1-Score	FPS (CPU)
YOLOv8	80%	0.78	~15 FPS
YOLOv5	72%	0.70	~12 FPS
YOLO-NAS	75%	0.73	~10 FPS
RT-DETR	68%	0.67	~7 FPS

The end