UNIVERSITÀ DEGLI STUDI DI TRIESTE

Detection, classification and tracking of fishes: an analysis of the YOLOv8 model

July 25, 2025

Fish detection problem

History of fish detection

YOLO

YOLOv8

Sub-things to explain

Datasets

Configuration files

```yaml
train: path_to_train/images
val: path_to_val/images
test: path_to_test/images

nc: 1
names: ['Fish']
```

```yaml
train: path_to_train/images
val: path_to_val/images
test: path_to_test/images

nc: 23
names: [
  'Caranx_sexfasciatus',
  'F1',
  'F2',
  'F3',
  'F4',
  'F5',
  'F6',
  'F7',
  'Acanthopagrus_palmaris',
  'Lutjanus_russellii',
  'acanthopagrus_and_caranx',
  'acanthopagrus_palmaris',
  'Gerres',
  'Caranx',
  'Amniataba_caudivittatus',
  'gerres_2',
  'gerres',
  'Epinephelus',
  'Fish',
  'juvenile',
  'palmaris',
  'EJP',
  'caudivittatus'
]
```

YOLOv8 models

Training

Inference

The Tracking Challenge: Occlusion & ID Switches

Problem: In complex underwater scenes, fish are frequently occluded by objects or other fish.

A naive tracker will often:

- **Lose the track** entirely when the fish disappears.
- **Create a new ID** when the fish reappears (ID Switch).

Goal: Maintain a persistent, correct ID for each fish despite occlusions.

The ByteTrack Algorithm

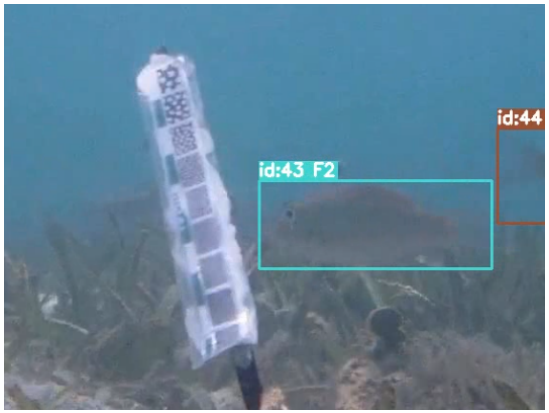
Stage 1: High-Confidence Match

- Uses motion prediction (Kalman Filter) to estimate the next position of existing tracks.
- Matches these predictions with high-confidence detections from the YOLO model.

Stage 2: Low-Confidence Match

- Instead of discarding low-confidence detections, it keeps them.
- It attempts to match these "doubtful" detections to the tracks that were lost in Stage 1.

Occlusion Example

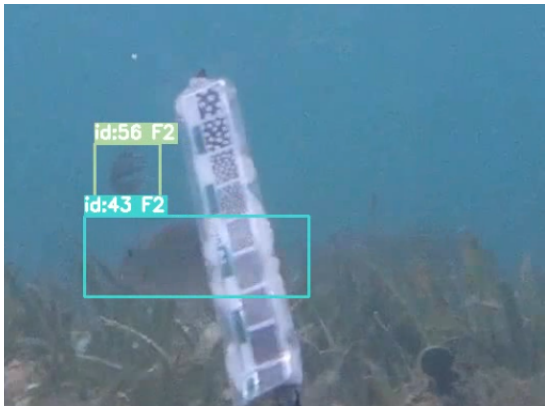


1. Before Occlusion

The fish ID 43 is clearly visible and being tracked.

2. During Occlusion

Fish ID 43 is now hidden behind an object.



3. After Occlusion

The fish reappears. ByteTrack re-identifies it as ID 43.