🐟 Enhancing Underwater Fish Tracking through Ensemble Methods and Autonomous Reinitialization

A robust semi-autonomous fish tracking framework for underwater video using OpenCV legacy trackers, Kalman filters, Mahalanobis distance filtering, and Covariance Intersection (CI) fusion. Includes support for both manual and automatic tracker reinitialization.

📂 Project Structure

File	Description
main.py	Main tracking pipeline: trackers + Kalman + CI fusion
mark_reinit.py	Manual tool for GUI-based reinitialization annotation
test_label.py	Visualizer for ground truth label overlays
requirements.txt	Python dependencies
data/	Output data folders (CSV + Evaluation python-based file)
labels/	Ground-truth data folders (YOLO-style format)
reinitialization_data/	Folder to store annotated or auto-generated reinit data

⚙️ Installation

Set up your environment and install dependencies:

```
git clone https://github.com/benztimm/thesis_fishtracking.git
cd thesis_fishtracking
pip install -r requirements.txt
```

▶ Usage

Main Tracking Script

```
python main.py
```

Press x during playback to select an ROI and reinitialize trackers. Press Esc to quit.

Evaluation variable:

Variable name	Description
TRACKER_NUM	configure number of tracker run in the program
COLLECT_DATA	True: Save output to CSV, False: not save
SWITCH	`kalman` or `tracker` mode
VERSION	Version/iteration name of output file
DATA_FOLDER	data folder name (e.g. 'data_9862')
VIDEO_NAME	Video file name (e.g. video9862.mp4)
LABEL_FOLDER	Ground-truth folder name
OUTPUT_WRITE_VIDEO	True: Save video output, False: not save

Annotate Reinitialization Points

```
python mark_reinit.py
```

Variable name	Description
video_name	File name of the input video
label_folder	Ground-truth folder name
reinitialization_data_folder	Output folder name
reinitialization_data_filename	Output file name

- Press x to select a new bounding box and pick trackers via GUI.
- Saves data into reinitialization_data/reinitialization_data_*.csv

Visualize Ground Truth

```
python test_label.py
```

Variable

Variable name	Description
video_name	File name of the input video
label_folder	Ground-truth folder name

Overlays label bounding boxes from labels_(video) folders onto video frames.

📈 Evaluation

Metrics used:

- RMSE (Root Mean Squared Error)
- Euclidean Distance
- IoU (Intersection over Union)
- Inside Ratio: % of frames where tracker's centroid is inside GT box
- Reinitialization Count and Failure Count

Data Folder

📊 Tracker Evaluation and Visualization

This directory contains evaluation logs and visualization tools for comparing tracking performance between raw outputs and Kalman-enhanced fusion on a specific underwater video dataset.

🧩 Contents

File	Description
evaluate.py	Prints summary statistics for all trackers and fused output
plot_distance_overtime.py	Line plot of Euclidean distance over time for each tracker
plot_rmse_overtime.py	Line plot of RMSE per frame per tracker
plot_iou_overtime.py	Line plot of IoU per frame for fused tracker (or optionally all)
*.csv	Evaluation results generated from the main tracking pipeline

📌 Evaluate Summary Statistics

```
python eval.py
```

This script will:
- Load metrics from both `tracker` and `kalman` versions
- Print:
  - RMSE
  - Max/Min/Avg distance
  - Inside ratio
  - Reinitialization count
  - Failure count
  - Mean IoU per tracker

📈 Plot Distance Over Time

```
python plot_distance_overtime.py
```

Generates `distance_over_time_<option>_<version>_<title>.png`, a graph showing per-frame Euclidean distance.

📉 Plot RMSE Over Time

```
python plot_rmse_overtime.py
```

Generates `rmse_over_time_<option>_<version>_<title>.png`, a graph showing root mean square error over frames for each tracker.

🧠 Plot IoU Over Time

```
python plot_iou_overtime.py
```

Generates `iou_over_time_<option>_<version>_<title>.png`, a graph showing intersection over union per frames for each tracker.