Fish detection and tracking – instructions

In order to train and evaluate the algorithm, you should complete the following steps:

1) Create training data
2) Train the model
3) Run algorithm on video
4) Annotate test video
5) Analyze results

Python packages required are:
easydict, matplotlib, opencv-python, pillow, scipy, tensorflow.

Create training data

The goal of this step is to create training images to train the model. We take a video, cut it to frames at one second intervals, and use them for the training. This is done by running the script extract_images_from_video.py.
You should to setup the following parameters:

```
REDUCE_RATE_FLAG = True  # True to reduce capture rate. False to use original rate
CAPTURE_INTERVAL_SECONDS = 1  # Capture every second
CUT_ROI_FLAG = True # If true the video is truncated around the ROI
roi = {'top left xy': (900, 75), 'bottom right xy': (2000, 1175)}  # Region Of Interest in video
video_path = r"C:\Users\d_kip\work\lab_projects\fish_Noise\Video" # Path of the mp4 video file
video_file = 'VIDEO_20230304_100716744.mp4'  # Name of video file
```

Note: roi is the region of interest in the video where the fish present, it is defined by its top-left and bottom-right corners, and should be square. The roi can change if the camera's position changes, and it may be required to fine-tune it's boundaries according to the specific video file.

Train the model

Training is done using the colab notebook: Few Shot ROI Fish Detection_TF_lite3.ipynb. A copy of this notebook is in: https://github.com/drorki/fish_tracking.git

You should specify the path for training and test images (test images are only for illustration):

```
test_image_dir='/content/drive/MyDrive/Fish_Noise/VIDEO_20230223_133606599_1FPS_ROI/"
```

```
train_image_dir='/content/drive/MyDrive/Fish_Noise/VIDEO_20230304_100716744_1FPS_ROI/'
```

On the training stage you can use the colab annotation tool to annotate the data, or use the pre-annotated data in gt_boxes by comment/uncomment the code sections.
The output of the notebook is a model that you can download and use as a fish detector for the tracking algorithm.
The test stage is only for illustration. You can prepare your test data following the same procedure you used for the training set.

Run algorithm on video

This is done by running object_tracker_tflite.py.
This script is originally based on: https://github.com/theAIGuysCode/yolov4-deepsort. However, there are many changes, and a main change is that the detection model is the one you trained (instead of YOLO in the original code).
You should download your trained model into './data' folder. Then set the model name in the PARAMETERS section of object_tracker_tflite.py, e.g,:
model_path = 'data/model7_ROI.tflite'

Annotate test video

You should follow the procedure on the file data_annotation.pdf

Analyze results

There are two scripts:

1. analyze_tracking.py – analyze tracking for a specific annotated fish
2. analyze_detection.py – evaluate the total detection performance, and tracking coverage for each fish

Running analyze_tracking.py
On the PARAMETERS section, set annotations_path to the directory that contain the annotation files, and filename2load to the pkl file containing tracking results, e.g.:
filename2load = 'tracks_20230302-122106.pkl'
the parameter dist_thd_pixels is the number of pixels allowed around each annotated track for a detection to be considered as a part of the track.

Running analyze_detection.py
On the PARAMETERS section, set the list annotation_paths to contain all the paths of the annotations. Similar to the procedure above, the variable filename2load should be the pkl file containing tracking results.