Initial Project Setup – CGS-PIEJ Eye Tracker ML

1. Clone the repository:

- https://github.com/Saul-Caban-UPRM/CGS-PIEJ-Eye-Tracker-ML.git

2. Install Python 3.12

- Download and install Python 3.12 from the official site Python 3.12.0
- Make sure to check "Add Python to PATH" during installation.

3. Upgrade pip

- In terminal: pip install --upgrade pip

4. Install Python dependencies

- In terminal: pip install ultralytics
- In terminal: pip install opencv-python

YOLO Training and Labeling Guide – CGS-PIEJ Eye Tracker ML

1. Prepare your video

- Obtain one of the videos from the eye tracker.
- Keep the video filenames consistent (for easier tracking).
- Place the video in the videos/ folder of the project.

2. Convert video to frames

- Run the video-to-frames script located in the utils/ folder.
- python utils/video_to_frames.py
- After running, you should see frames saved inside the screenshots/ folder.

3. Review frames

- Manually go through each frame to select the items you want to label.
- Example: a pen.
- Make a note of which items you want to track consistently across all images.

4. Label images using Roboflow

- Go to https://roboflow.com/ sign up and create a new project.
- Upload the selected frames.
- Label each item you want to track.
- Create custom labels.
- Ensure label names are consistent with your project item names.
- Once labeling is complete, download the dataset as a ZIP file.
- Make sure to have select YOLOv8 as annotation format

5. Prepare YOLO training data

- Extract the downloaded ZIP file.
- Copy the train folder into the project at:
- yolov8_training/train
- Copy data.yaml from the extracted ZIP into yolov8_training/.

- Open data.yaml and make sure the paths are set correctly:
  - rain: ../train/images
  - val: ../train/images
  - test: ../test/images

6. Train YOLOv8 model

- Open a terminal inside the yolov8_training/ folder.
- Run the training command:
- yolo train model=yolov8n.pt data=data.yaml epochs=25 imgsz=320
  - epochs: number of learning cycles (how many times the model sees all images).
  - imgsz: image size for training (increase if your machine can handle it).
- After training completes, the model weights should be saved at:
- yolov8_training/runs/detect/train/weights/best.pt

7. Test the trained model

- Open main.py.
- Make sure this line exists (pointing to your trained model):
- MODEL_PATH = Path("yolov8_training/runs/detect/train/weights/best.pt")
- Run the script and observe results on your videos.

In the project theres a video guide going briefly over these steps.