**Please use this file to understand the use of each file**

This document provides a brief explanation of the purpose and usage of each notebook and Python script included in the submission artifact for the *Real-Time Fall Detection System using YOLOv11* project.

**1. Fall\_Detection\_Modeling\_using\_YOLOV11.ptnb**

Main Jupyter notebook used for:

* Preprocessing dataset (initial overview)
* Training the YOLOv11 model
* Evaluating training results
* Exporting the best model for deployment

**2. caucafall\_dataset\_structuring.ipynb**

* Jupyter notebook created to structure the CAUCAFall dataset.
* Reorganizes images and annotations from multiple folders into a consistent YOLO-compatible format.
* Splits the dataset into train, val, and test folders.

**3. invert\_falls\_label.py**

* Python script that corrects labelling issues in the LE2I augmented dataset.
* Inverts the incorrect fall label from "0" to "1" to ensure consistency across all datasets.

**4. balanced\_augmentation\_utils.py**

* Used to apply advanced augmentation techniques to balance the dataset classes.
* Performs transformations like flipping, rotation, blur, brightness, etc., primarily to increase fall instances.

**5. advanced\_train\_yolo.py**

* Standalone training script used for full-scale model training on the structured and augmented dataset..

**6. hard\_sample\_finetuning\_on\_train23.py**

* Script for fine-tuning the best-performing model (train23) using hard samples (images where the model performed poorly).
* Improves model robustness and recall.

**7. false\_prediction\_finder.py**

* Analyses predictions to locate false positives and false negatives.
* Helps identify hard samples that can be used for fine-tuning.

**8. model\_comparison.py**

* Compares outputs from two models (train23 and fine-tuned) on the same test images.
* Draws bounding boxes and confidence scores for each model, helping in visual evaluation.

**9. Falldetection\_web/**

* Django-based folder containing the complete web system code.
* Includes frontend (HTML/CSS), backend (Python Django), and WebSocket logic for real-time RTSP video analysis and fall detection.