A. Person Identity Catalogue (cross-clip)

I began by extracting key frames from each clip to reduce redundancy. I implemented 3 methods and tested the tracking performance for each:

- Regular frames extraction (all frames)
- Top-k method that selected the top K frames showing significant visual change, defined as frames where the pixel-wise difference.
- Adaptive thresholding method that selected frames showing significant visual change, defined as frames where the pixel-wise difference exceeded the mean + k × std.

Next, I used YOLO v11n (yolo11n-seg) for person segmentation (class 0) with several configurations, and relied on YOLO's built-in tracking mechanism to assign consistent Local IDs to each person across frames within each clip. For each track, I cropped the detected person and stored the cropped image as well as the masked image (black background outside the segmentation), which allowed me to build a dataset of per-person appearances and tracks in each clip.

Notes, Limitations & Possible Improvements:

• Heavy occlusions or low-quality frames occasionally produced duplicate LIDs.

To extract Re-Identification embeddings (normalized), I unified local identities across clips, I used the Torchreid library with the pretrained OSNet x1.0 model to for each cropped person image.

I then clustered all embeddings jointly using a density-based algorithm (OPTICS or DBSCAN) with cosine distance. None of them results in good managed clusters so I decided to implement a logic mechanism for unify local IDS across different clips:

The resulting clusters represented global IDs (GIDs) shared across all clips.

Finally, I grouped each local track by its most frequent cluster assignment to ensure one-to-one mapping between LID and GID. Visualization utilities displayed example crops per cluster to validate the grouping qualitatively.

- Embeddings were computed using pretrained weights only; fine-tuning could improve domain adaptation.
- Low resources (time and GPU) would probably allow me to test transformer-based ReID models or integrate extra features for improved discrimination.

B. Scene Classification

My approach to the scene classification task was based on combining CLIP for visual captioning with a lightweight LLM for semantic reasoning over the generated descriptions.
The intended pipeline extracted textual summaries from representative frames and then prompted the LLM to classify each clip as “normal” or “crime” based on visual cues.

However, due to high computational demands and repeated runtime failures that resulted in excessive CPU consumption, the pipeline could not be executed successfully to completion.
The obtained results were inconsistent and below expectations, and therefore I chose not to include this component in detailed discussion within this brief.
I acknowledge these limitations and their potential implications for the overall outcome.