

Problem Statement 2

Use Case Title:

Real-Time Individual Detection from CCTV Feeds

Problem Statement

Given real-time streaming from CCTV cameras, the goal is to accurately detect and match individuals across different time frames and camera locations. The solution should be highly accurate to ensure minimal false positives and false negatives.

Core Requirements

1. Real-Time Streaming Integration:

- Stream live footage from multiple CCTV sources.
- Efficiently process frames in real-time.

2. Face Detection and Matching:

- Detect individual faces in real-time with high accuracy.
- Use feature extraction (like facial landmarks) for robust matching.
- Support for multi-angle detection (side profiles, partial occlusion).

3. Cross-Camera Matching:

- Recognize the same individual across different camera views.
- Handle varying lighting conditions and background changes.

4. Scalability and Latency:

- Solution should scale to multiple camera feeds.
- Maintain low latency for real-time decision-making.

5. Data Privacy and Security:

- Ensure secure handling of video streams.
- Anonymize data if required for compliance.

Tech Stack Recommendations

1. Real-Time Streaming:

- Kafka, Pulsar, or WebRTC for low-latency streaming.
- GStreamer for multimedia processing.

2. Object Detection & Matching:

- YOLOv8 or RetinaNet for face detection.
- FaceNet, DeepFace, or ArcFace for facial recognition.
- DLIB or Mediapipe for facial landmark detection.

3. Distributed Processing:

- Apache Flink or Spark Streaming for parallel processing.
- Kubernetes for container orchestration if scaling is needed.

4. Database:

- Redis or Cassandra for fast lookup of face embeddings.
- TimescaleDB or InfluxDB for time-series data of events.

5. Visualization:

- Streamlit or Flask for real-time monitoring dashboards.

Evaluation Metrics

1. Detection Accuracy: Precision, Recall, and F1-Score.
2. Latency: Time taken to process and detect an individual.
3. Cross-Camera Match Rate: Ability to recognize the same person across feeds.
4. Scalability Performance: Handling multiple streams concurrently.

Hackathon Deliverables

1. A working prototype with live camera feed integration.
2. Real-time detection with visual indicators.
3. Cross-camera matching of individuals.
4. Dashboard for monitoring detected individuals and match confidence.