Idea:

1. The setup consists of multiple static CCTV IP cameras located at strategic positions. Each camera has an IP address with which it is referred by. An **IP table** keeps a record of all the camera IPs, thereby keeping track of **camera neighbours.**

2. Server will contain a database consisting of Person identification details , consisting of faces that will be trained on the **Local Binary Patterns Histogram** algorithm, which is ideal for Real time Recognition.

3. *Face Detection* using **Histogram of Oriented Gradients (HOG)** and **Haar cascades in OpenCV.**

4. *Recognition* algorithms will be run on each frame relayed by the IP cameras. On recognizing a **POI**(Person of interest), the module will begin tracking the individual(s) using **GOTURN tracking** API, on top of the system's *GPU* to enhance the computation efficiency.

5. The camera relaying this information to the server will have a check on the **boundaries** of the recognized individual(s), thereby enabling us to know when the POI moves out of it's frame boundaries.

6. The **IP table** is *referenced* to determine which of the IP cameras acts as neighbour, to enable continuous tracking of the POI.

7. The camera sends the person details along with the last known position to the server. The server then triggers an alert to the neighbour camera (B) and this makes B take over the tracking of the required individual.

8. When **multiple-people** tracking is required, we use **multithreading**, letting each thread track each POI individually, relaying camera information from the server, simultaneously. Since multiple cameras capture video simultaneously, and each camera can track multiple people within its frame, we thus enable *multi-person tracking over the entire region of interest*.

9. Output through a monitor with a **user interface** for automatic viewing of the final relayed *tagged feed*.
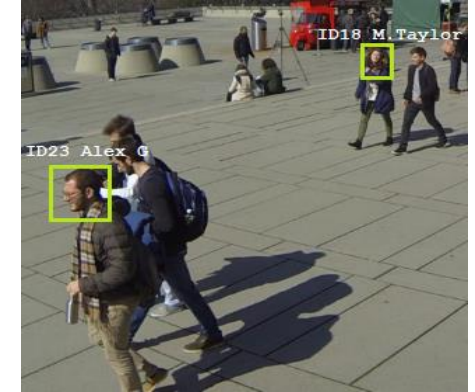
Input from CCTV Camera Feed
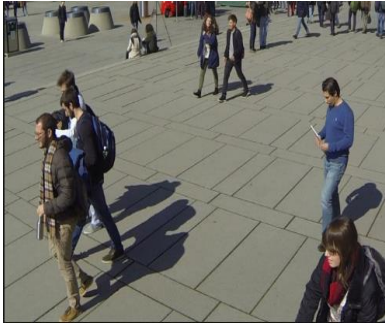
Detection of pedestrians using **Histogram of Oriented Gradients(HOG)** and **Haar Cascade** using OpenCV
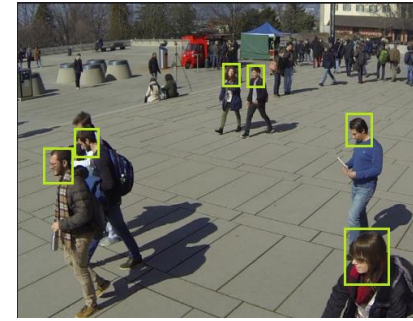
Recognition of Person/s of Interest(POI) By **Local Binary Patterns Histogram** Algorithm

Tracking of POI by **GOTURN Tracking** using system GPU

Check CCTV frame limits – POI **exits frame** of view of current camera

Obtain neighbour's **IP address** from server database

IP camera feed switching

Trigger alert to neighbouring CCTV camera through IP

Neighbouring CCTV feed **takes over** tracking of POI

Send Tracking data to Server For data storage and further processing

Send Tracking data to Server For data storage and further processing

Technology Stack

CCTV IP Camera – Input Video Feed

For Face Detection
Histogram of Gradients (HOG) and
Haar Cascade Using OpenCV

Local Binary Pattern Histogram Algorithm for Multi-person Face Recognition

GOTURN Tracking of recognized individual(s) over multiple frames

Server For Storing IP Addresses of all static IP Cameras using *IP Table for switching of cameras*

Output Monitor with user interface for automatic viewing

Use Cases:
- For the purpose of surveillance at the scene of a crime, to follow the perpetrator.
- CCTV Security surveillance measure for households or corporations.

Showstoppers:
- We have already developed a project that included real-time face recognition of video feed provided by an Android phone. We developed an App that converts the phone into an IP camera, relaying information to a particular IP address on the same network.
- In commercial specific use, the images from the identification card of employees can be trained.
- Linking of Aadhaar details to the recognition of pedestrians to enable complete identification.
- The entire flow of control is pipelined and would work seamlessly in real-time.

Future Enhancements:
- If the person of interest uses a vehicle for transport, then vehicle license plate detection can be integrated to obtain more detailed information as well.
- The clarity of the IP cameras could be improved by using the cameras from discarded phones, thereby enhancing the quality of imaging whilst simultaneously reducing the costs.