# Introduction

This project aims to develop a computer vision-based pedestrian detection system for videos. The system will be trained to detect pedestrians within a specified region of interest (ROI) in the video. The goal of this project is to accurately and efficiently detect pedestrians in real-time, providing a useful tool for various applications such as surveillance and autonomous vehicles. The project will involve the use of state-of-the-art deep learning techniques and machine learning algorithms to analyze the video data. The system will be trained and tested on a dataset of video footage containing pedestrians in various scenarios. The project will be executed in several stages, starting with research on current pedestrian detection methods, followed by the development and testing of the system. The final outcome of the project is a working pedestrian detection system that can be integrated into a real-world application. Additionally, the system should also be robust to variations in lighting, angles, and other environmental conditions. The project also aims to optimize the system's computational efficiency, making it suitable for real-time use in resource-constrained environments.

Objective

The objective of this project is to design and implement a computer vision-based pedestrian detection system for videos.

The system will be able to

* take a test video and a bounding box in a Json file(top left, top right, bottom left, bottom right) as input.
* perform detection of pedestrians and capture their images if they fall in the region of interest.
* Redact the faces of people that are not in the ROI.
* Track pedestrians and determine where they left from the region of interest(ROI).

# Model Used

YOLOv7 is a real-time object detection model that is based on the YOLO (You Only Look Once) architecture. YOLO is a single-stage object detector, meaning it processes the entire image only once to detect objects. It is known for its speed and accuracy, making it well-suited for real-time object detection applications.

YOLOv7 is the latest version of the YOLO architecture, and it has several improvements over previous versions. One of the main improvements is the use of a more efficient backbone network, which allows for faster object detection. Additionally, YOLOv7 incorporates several new techniques such as Mosaic data augmentation, and CIoU (complete Intersection over Union) loss function to improve accuracy.

YOLOv7 is pre-trained on COCO (Common Objects in Context) dataset and can detect 80 different object classes such as person, car, bike, etc. It can run at high frame-rate and can be used for real-time object detection applications such as surveillance, autonomous vehicles, and video analytics.

# Approach

* Approach to take a test video and a bounding box in a Json file(top left, top right, bottom left, bottom right) as input

There are several approaches that can be taken to process a test video and a bounding box in a JSON file as input for object detection. One approach would be to use a pre-trained object detection model such as YOLOv7, and use it to detect objects in the test video. The JSON file containing the bounding box coordinates can be used to define the region of interest (ROI) for the object detection.

The first step would be to read the JSON file and extract the bounding box coordinates. Then, the test video would be processed frame by frame, and each frame would be passed through the object detection model. The output of the model would be a set of bounding boxes and class labels for each detected object. The coordinates of the bounding box in the JSON file can be compared with the output of the model to verify if the object of interest is detected within the specified ROI.

Another approach would be to use the bounding box coordinates to crop the ROI from the test video, and then pass the cropped ROI through the object detection model. This can be more efficient since it reduces the amount of data that needs to be processed by the model.

Finally, the output of the model can be visualized by drawing the bounding boxes on top of the original video frames, and display the class labels for the detected objects. The output can also be saved in a different video format or in a JSON file for further analysis.

# Approach to perform detection of pedestrians and capture their images if they fall in the region of interest

The approach to perform detection of pedestrians and capture their images if they fall in the region of interest can involve several steps:

1. Acquire a dataset of images that contain pedestrians in the region of interest.
2. Use a object detection model, such as YOLO or Faster R-CNN, to train on the dataset and detect pedestrians in new images.
3. Set up a camera or video feed in the region of interest and use the trained object detection model to detect pedestrians in the live feed.
4. When a pedestrian is detected within the region of interest, capture an image of the pedestrian.
5. Store the captured image for further processing or analysis.

It is important to note that this task may require some knowledge of computer vision and machine learning, and that the specific implementation details may vary depending on the specific use case

* Approach to redact the faces of people that are not in the region of interest (ROI).

The approach to redact the faces of people that are not in the region of interest (ROI) can involve several steps:

Use a face detection model, such as MTCNN or OpenCV's Haar cascades, to detect faces in the image.

Use a facial recognition model, such as VGGFace2 or FaceNet, to compare the detected faces to a database of known individuals or a pre-defined list of individuals that are in the ROI.

For any faces that are not recognized as being in the ROI, use image processing techniques such as pixelation or blurring to obscure the face.

Save the redacted image for further processing or analysis.

It is important to note that this task may require some knowledge of computer vision and machine learning, and that the specific implementation details may vary depending on the specific use case. Additionally, it is important to consider the privacy and legal implications of redacting faces in images.

* Approach to Track pedestrians and determine where they left from the region of interest(ROI).

The approach to track pedestrians and determine where they left from the region of interest (ROI) can involve several steps:

1. Use a object detection model, such as YOLO or Faster R-CNN, to detect pedestrians in live video feed from cameras or drones within the ROI.
2. Use a multiple object tracking algorithm, such as Kalman filter or Deep SORT, to assign unique IDs to each pedestrian and track their movements within the ROI.
3. Use the position and movement information of the pedestrians to determine where they entered and left the ROI. This can be done by comparing the positions of the tracked pedestrians to a pre-defined map of the ROI and looking for changes in direction or speed that indicate entry or exit points.
4. Store the entry and exit points along with the unique IDs of the pedestrians for further processing or analysis.

It is important to note that this task may require some knowledge of computer vision and machine learning, and that the specific implementation details may vary depending on the specific use case and the quality of the image and video feed. Additionally, it is important to consider the privacy and legal implications of tracking individuals in public spaces.

# Summary

The project aims to detect persons in a CCTV camera video and capture their images if they fall within a predefined region of interest (ROI). The project will involve several steps:

1. Acquiring a dataset of images that contain persons in the ROI.
2. Using a object detection model, such as YOLO or Faster R-CNN, to train on the dataset and detect persons in new images.
3. Setting up a CCTV camera or video feed in the ROI and using the trained object detection model to detect persons in the live feed.
4. When a person is detected within the ROI, capturing an image of the person.
5. Storing the captured image for further processing or analysis.

The project will require some knowledge of computer vision and machine learning and the specific implementation details may vary depending on the specific use case. Additionally, it is important to consider the privacy and legal implications of capturing images of individuals in public spaces.