Name Avinash Jadhav
Contact: 8888690633
Email id: avijadhavak@gmail.com

Solution

Problem Statement
Design and develop an object detection and tracking algorithm to count the number of people entering and exiting a premise from a given video feed. The algorithm should accurately identify individuals, track their movements, and maintain an entry and exit count. The result should display the entry and exit counts on the video screen.

Done with Requirements:

1. **Object Detection and Tracking Algorithm**
   - Successfully detected the object accurately using the YOLOv8 model.
   - Successfully tracked the object and assigned a unique ID.
2. **Counting Logic**
   - Implemented the counting logic in code and accurately counted the entry and exit counts. Please refer to the output videos in the output_videos folder: enter_example.avi and exit_example.avi.
3. **FPS Optimization**
   - Currently, the detection FPS is between 3 and 5 FPS, which is displayed in the output video frames. We can change the frames reading FPS and frame size. I used a frame size of 1020x500 because it gave the best results for me and was convenient for my hardware. I can do more optimization in the FPS code.
4. **GitHub Repository**

Created a repository named peopleCounting with the following directory structure:

- output_videos: Where the final result is stored. I have provided both features in the code: writing and not writing the video. You only need to uncomment the code to enable/disable this feature.
- input_videos: Where the input video is given.
- models: Where the actual YOLOv8m model is stored.
- cords.py: Where the area1 and area2 coordinates are stored.
- mains.py: Where the main code is present. To run the code, type python3 mains.py.
- tracker: Where the tracker logic is present. I used the tracker class in the main file.

Approach:

1. Analyze the video to identify the paths that people take when entering and exiting the office area.
2. Use the mouseOver cv2 method to get the coordinates of the entrance and exit areas.
3. Draw two rectangles on the video feed, one for the entrance area and one for the exit area.
4. Use a person detection and tracking model to identify and track people in the video feed.
5. Maintain a list of the unique IDs of people who have passed through the entrance area.
6. When a person passes through the exit area, check if their unique ID is in the list. If it is, then the person is exiting the office area and the exit count is incremented.
7. Repeat steps 5 and 6 for all people in the video feed.

Code Explain

The code you provided is a Python implementation of a person counter. It uses the YOLOv8 object detection model to identify people in a video feed, and then uses a tracker to keep track of them as they move around. The counter also maintains two sets of IDs: one for people who are entering a room, and one for people who are exiting.

The code works by first loading the YOLOv8 model and the tracker. Then, it opens the video file and starts reading frames. For each frame, the code does the following:

1. Resizes the frame to a consistent size.
2. Passes the frame to the YOLOv8 model to detect people.
3. Updates the tracker with the detected people.
4. Check each tracked person to see if they are entering or exiting the room.
5. Draws bounding boxes around the tracked people and displays the number of people entering and exiting the room.

The code also calculates and displays the adjusted frame rate, which is the actual frame rate of the video divided by the time it takes to process each frame. This can be useful for debugging and optimizing the code.

Here is a more simplified explanation of the code:

1. The code starts by importing the necessary libraries: cv2, pandas, numpy, ultralytics, and tracker.
2. It then defines a class called PersonCounter. This class has the following attributes:
    - video_path: The path to the video file.
    - output_path: The path to the output video file.
    - model: The YOLOv8 object detection model.
    - area1: A list of four points that define the entrance area.
    - area2: A list of four points that define the exit area.
    - class_list: A list of all the classes that the YOLOv8 model can detect.
    - tracker: A tracker object for tracking people.
    - entering: A set of ID numbers for people who are entering the room.
    - exiting: A set of ID numbers for people who are exiting the room.
    - people_enter: A dictionary that maps ID numbers to the (x, y) coordinates of the people who are entering the room.
    - people_exiting: A dictionary that maps ID numbers to the (x, y) coordinates of the people who are exiting the room.
    - target_fps: The target frame rate.
    - frame_counter: A counter for the number of frames that have been processed.
    - start_time: The time at which the code started processing the video.
3. The PersonCounter class also has the following methods:
    - load_class_list(): Loads the class list from a file.
    - process_frame(): Processes a single frame of video. This method detects people in the frame, updates the tracker with the detected people, checks each tracked person to see if they are entering or exiting the room, and draws bounding boxes around the tracked people.
    - run(): Runs the person counter. This method reads frames from the video file, processes each frame using the process_frame() method, and writes the processed frames to the output video file.

The main function of the code creates a PersonCounter object and calls the run() method on it. This starts the person counter and processes the video file.
The code also has a number of other methods and variables, but the ones described above are the most important ones.

Steps To run the code

1 git clone .https://github.com/avi3108/peopleCounting.git
2. Download the requirements packages mentioned in requirements.txt
3.Put the input video in input_videos folder

2. Run the main file : python3 mains.py