

YOLOv9-based Real-time Human Detection, Tracking, and Directional Counting in Public Environments

Sesha Phaneendra G.¹, Shaik Affika Sultana²

^{1,2}Department of Computer Science and Engineering, Anantha Lakshmi Institute of Technology & Sciences, JNTUA, Ananthapuramu, India.

E-mail: ¹seshaphanee@gmail.com, ²shaikafi1410@gmail.com

Abstract

A real-time people-counting system for crowded scenarios needs to have effective human detection, temporal tracking, and estimation of directional movement despite possible occlusions and different scenarios. This research work proposes a real-time people-counting system that utilizes the YOLOv9 algorithm for video-frame processing, Non-Maximum Suppression (NMS), centroid tracking, and virtual line-crossing for the estimation of IN/OUT direction. This study uses the CrowdHuman annotated image dataset, which is split into training, validation, and testing datasets. Human instances detected in the video are localized by bounding boxes and linked with the centroids' positions between the consecutive frames, and the crossing points are detected by crossing of centroid positions on a certain virtual border. This pipeline can process both pre-recorded videos and live webcam feeds. The proposed mode has resulted in an accuracy of 96%, 91% precision, and 91% recall.

Keywords: YOLOv9, Human Detection, Centroid-Based Tracking, Virtual Line Crossing, Non-Maximum Suppression, CrowdHuman Dataset, Real-time Video Analytics.

1. Introduction

Applications such as the real-time detection and counting of humans play an important role in using computer vision for security-related purposes. Real-time automated vision systems can analyze video streams to detect humans and their movements and therefore lessen the need for human surveillance. Counting humans in real-time is difficult due to several factors such

as occlusions, overlap between humans, changing light conditions, complex backgrounds, and varying camera views.

Deep Learning-driven object detection technology has been successful in human detection tasks due to the ability to classify and locate objects at the same time in a frame in a video. Out of the many techniques available, YOLO models are ideal for real-time detection applications because of their single-shot detection process [6], [7].

In continuous counting, object tracking needs to be incorporated into frame level detection for enabling a correspondence between objects through frames. Tracking-based techniques can help to minimize counting duplication and provide directionality analysis capability. Recent works have shown that such approaches are feasible for real-time crowd monitoring and people counting applications [8], [9], [10].

The research introduces a real-time human detection and counting algorithm, which uses YOLOv9, Centroid tracking, and Virtual line crossing for detecting, tracking and counting humans. The method supports both offline and online processing, where the online part consists of the processing of real-time webcam feed. The individuals are tracked between the consecutive frames, and based on their passing through a virtual line, the number of IN and OUT persons is calculated.

2. Related Works

People counting using vision-based technology has been extensively studied in applications such as surveillance and crowd monitoring. The initial studies in this field involved bidirectional counting and video detection, proving that it is possible to detect people coming in and moving out of the monitored areas [1]. Vision-based technology has also been used in transportation systems where camera detection was used to determine the occupancy rate of passengers inside buses [2]. People counting using image processing methods using MATLAB software has also been considered [3].

The indoor monitoring systems later made use of vision-based methods to calculate the number of people in the monitored area [4]. This was applied to attendance monitoring purposes, where vision-based methods have been used in the calculation of human presence [5]. Recent studies have been carried out on more accurate and efficient real-time counting by using

cascaded object detection techniques [6].

The emergence of improved YOLO architectures has also improved real-time object detection. In the case of YOLOv9, there is programmable gradient information that improves the feature learning and detection process [7]. Many deep learning approaches have also been considered for crowd counting problems; recent studies have considered the ability and difficulties faced in the application of these approaches in various crowd conditions [8]. In one case, YOLO detection with KCF tracking was used to perform crowd counting, thus combining the two processes [9]. In recent times, real-time people tracking and counting with overhead cameras were performed using AI detection and tracking [10].

Overall, existing research shows the development from traditional vision-based people counting towards deep learning-based people detection and tracking. Nevertheless, issues of occlusions, dense crowd situations, viewpoint differences, and real-time people counting are still essential factors. This study is aimed at solving the problem of YOLO based detection and tracking along with line crossing for real-time people counting.

3. Proposed Methodology

The proposed system is designed as an end-to-end video-based detection and counting process for humans in public places. The system takes either pre-recorded video or live webcam feed as an input and analyses the input video frame-by-frame. Preprocessing is applied to each frame before it is fed to the YOLOv9 detector module to detect humans and draw bounding boxes around them. Non-Maximum Suppression (NMS) is used to get rid of overlapping detections.

The detections that are made are then fed to a tracking module using the concept of a centroid to ensure that the individuals are traced consistently from one frame to another. The position of the centroids is constantly checked relative to a counting line that has been virtually defined. As an individual move across the counting line, the direction is detected, and the IN or OUT count is accordingly increased. The processed frames are then outputted using the system interface together with other parameters like detections, tracking information, direction of motion, and counts.

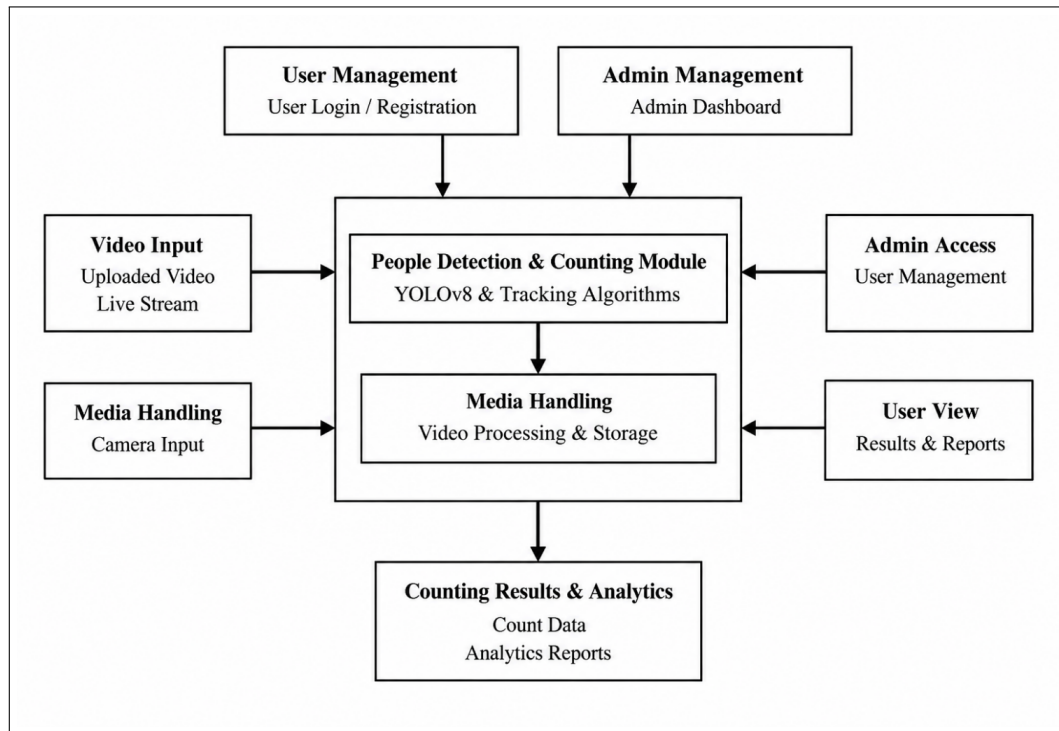


Figure 1. Proposed human detection and counting system architecture

3.1 Dataset and Input Acquisition

CrowdHuman dataset is employed for human detection and crowd analysis in the proposed research. The dataset comprises annotated images depicting instances of humans under varying crowd scenarios. As presented in Table 1, the utilized dataset in this paper has 15,000 images distributed as 70% for training (10,500 images), 15% for validation (2,250 images), and 15% for testing (2,250 images). Annotation in the dataset has been provided in terms of bounding boxes that span complete body, visible body, and head. This annotation allows localization of human instances and provides reference data for model evaluation [11].

In practical testing, the system can also accept recorded video files and web camera feed. The video input is segmented into frames and fed into the algorithm frame-by-frame. As a result, the system can detect and count people based on recorded and live video inputs.

The attributes of the dataset, such as the application, data type, distribution, annotation format, and key issue of dense crowd and occlusion of people, are also highlighted in Table 1.

Table 1. Characteristics of the crowdhuman dataset

Parameter	Description
Dataset	CrowdHuman
Application	Human detection and crowd analysis
Dataset type	Annotated image dataset
Training set	70% (10,500 images)
Validation set	15% (2,250 images)
Test set	15% (2,250 images)
Annotation type	Bounding boxes
Annotated regions	Full body, visible body, and head
Primary challenge	Dense crowds and human occlusion

3.2 Frame Preprocessing

The collected video will be analyzed frame by frame before the process of human detection. Resizing will be used first in order to have the same space representation of the frames used for the detection procedure. The normalization of the frames will then follow to ensure that all the frames are standardized, and the noise reduction process will be done to avoid any form of unnecessary variation of the visual information.

The preprocessed frames are fed one by one into the YOLOv9 detection module. Since the system processes the frame individually prior to performing any tracking operation, the order of the frames is maintained in the next phase of centroid tracking. The above processing method is used for both the uploaded video files and live webcam feed.

3.3 YOLOv9-based Human Detection

After pre-processing, every frame is passed to the YOLOv9 detector for localization of the instances of humans. The algorithm goes through the whole image and detects the people by creating bounding boxes around the recognized human instances. The output from the detection process serves as the spatial information for the next step of tracking and counting. YOLOv9 is chosen as the main detector of our system based on the comparison with YOLOv8 performed in the research.

For every human detected in the image, the information about the bounding box is considered in order to figure out their position within the frame. The detected humans become

the input for the following Non-Maximum Suppression step where the overlapping detection candidates are analyzed. Through NMS, the redundant bounding boxes are filtered out, and only the important detection remains for the individual.

Final human detections after the process of suppression are finally moved on to the tracking phase. Spatial coordinates related to each bounding box are utilized to determine the centroid that serves as the reference point for an individual to keep his/her movement consistent between consecutive frames. This means that the process of detection determines the locations of humans for further tracking and counting.

3.4 Centroid-based Tracking

After human detection and NMS, the surviving bounding boxes are then fed into the tracking process. The person that was detected is represented by the centroid of its respective bounding box. For a bounding box defined by the upper-left coordinates (x_1, y_1) and lower-right coordinates (x_2, y_2) , the centroid is calculated as:

$$C_x = \frac{x_1 + x_2}{2}, \quad C_y = \frac{y_1 + y_2}{2} \quad (1)$$

where C_x and C_y denote the horizontal and vertical centroid coordinates, respectively. After human detection and Non-Maximum Suppression, the surviving bounding boxes are then fed into the tracking process. The person that was detected is represented by the centroid of its respective bounding box.

3.5 Line-Crossing and IN/OUT Counting

The counting threshold is created by a horizontal line virtually drawn within the observed area. The vertical centroid location of the moving target is constantly checked against the predetermined location of the line Y_L . A The crossing occurs through checking the vertical locations of the targets in two successive frames

$$\begin{cases} C_y^{t-1} < Y_L \wedge C_y^t \geq Y_L \Rightarrow \text{IN} \\ C_y^{t-1} > Y_L \wedge C_y^t \leq Y_L \Rightarrow \text{OUT} \end{cases} \quad (2)$$

where C_y^{t-1} and C_y^t represent the vertical centroid positions in consecutive frames. After a tracked object crosses the virtual boundary line, its direction of motion is estimated from this position change along with incrementing the relevant IN or OUT counter. Tracking identity is preserved between frames in order to avoid re-counting in case of continuous motion. Directional people counting data is thus generated from tracked trajectories.

3.6 System Interface and Access Management

The designed software has an organized graphical interface that controls people counting processes and access management. The user module includes registration and login services, which allow authorized users to use available video processing capabilities. Once the users are authenticated, they have an option to select input sources for the chosen process, either uploaded video files or webcam videos.

The application has an additional feature of providing an administrative section. The administrator will have the ability to administer users, system settings, permissions, and overall system activity. It ensures that there is a division of roles of the user and administrator, thus, control of the application functionality.

4. Results and Discussion

4.1 Experimental Setup

The evaluation experiment was performed on the 15,000-images experimental subset of the CrowdHuman dataset, which consists of 10,500 training images, 2,250 validation images, and 2,250 test images, as described in Table 1 below. The dataset contains bounding box annotations of full-body, visible body, and head regions under crowd scenarios.

Implementation of the YOLOv9 model was done in the environment that uses DL with the Python programming language, using PyTorch and OpenCV. The training and evaluation can be done on the computer that has an Intel Core i7 processor, 16 GB RAM, and a GPU having 8 GB VRAM running 64-bit version of Windows 11 operating system. The Python 3.x environment with the capability to do DL with GPU was used for processing the model.

The model was trained using the training dataset, validated using the validation dataset, and tested using the independent test dataset. Accuracy, precision, and recall were used as metrics for evaluating the model. Besides, pre-recorded videos and real-time webcam feeds

were also used for evaluating detection, tracking, and direction-based IN/OUT counts of the integrated system.

4.2 YOLOv9 Detection Performance

The results obtained in terms of the quantitative performance of the YOLOv9 model can be seen in Table 2. With the 91% precision, it is clear that the human instances detected are actually humans since they are mostly accurately identified, thus having fewer false positives within the test scenes. With the 91% recall, the performance of the detector in terms of detecting most human instances available in the test samples, including those that arise in densely populated and occluded scenes depicted by the CrowdHuman data set, is captured. The fact that the precision and recall values are equal depicts an equal performance for both human instance identification and false positives.

The bounding boxes created from YOLOv9 are then provided for the centroid-tracking phase, where the location of the individual in each frame is tracked. Consequently, the performance achieved by the detection step, as shown in Table 2, will serve as an input for the next phases of tracking and directionality counting.

Table 2. YOLOv9 detection performance

Metric	YOLOv9
Accuracy	96%
Precision	91%
Recall	91%

4.3 Video-Based People Counting

Interface of the proposed system for video input is shown in Figure 2. The interface enables users to choose a recorded video file and then process it using the people-counting pipeline. The chosen video file becomes the input data for subsequent frame processing, people detection, centroid tracking, and directional counting.

This interface serves as the initial step towards loading the video data into the system before performing the detection and counting processes. Once the video is uploaded, it is then processed using the YOLOv9 algorithm in order to provide the results for human detection and

counting performed in subsequent stages of the system.

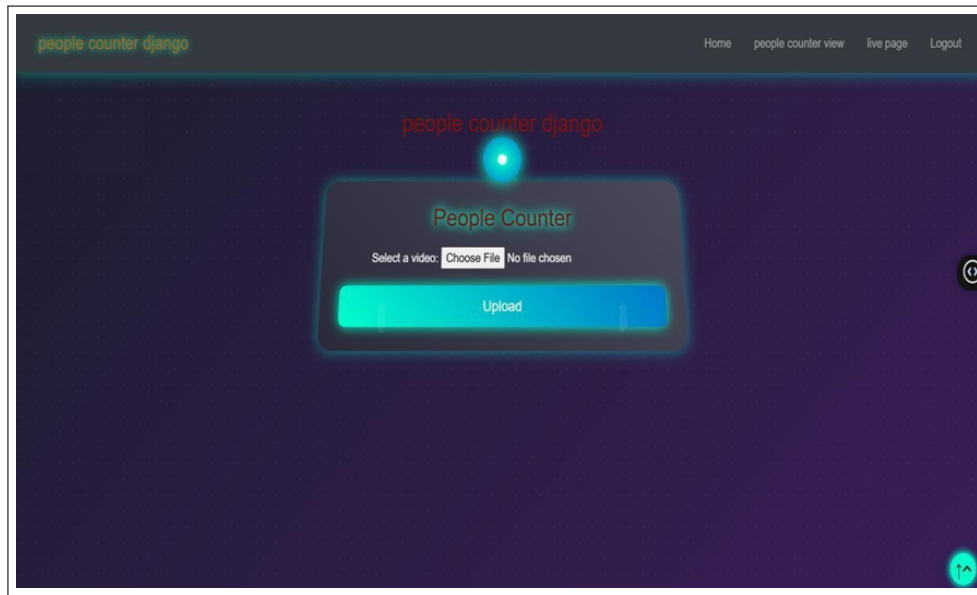


Figure 2. Video upload interface

4.4 Real-time Webcam-Based People Counting

The live implementation of the proposed model is depicted in Figure 3 through the use of a live feed from a webcam. The individual frames that are received are then analyzed using the YOLOv9 detector that has been trained to detect the presence of humans and provide bounding boxes for each one. The humans are tracked using their centroids.

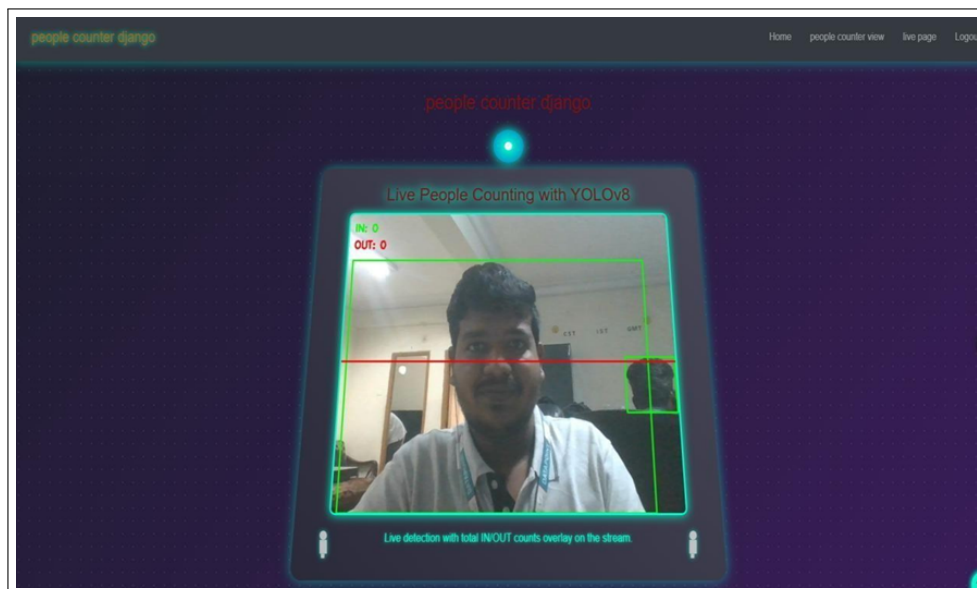


Figure 3. Real-time webcam people counting output

The information on the tracking process is compared to the previously set virtual counting line through continuous acquisition of video frames. After the movement of the centroid across the line boundary, the direction of the movement and the update of the count (IN or OUT) take place. Contrary to the previous case where the recording was completed prior to the experiment, the webcam system does detection, tracking, and counting in real time.

4.5 IN/OUT Counting Analysis

The performance of the proposed direction counting scheme is illustrated in Figure 4. In this system, the direction of travel of each detected person is determined by using the shifting of the centroid position in relation to the defined virtual counting line. If the detected person moves over the boundary in the defined direction of travel, then the IN count is increased.

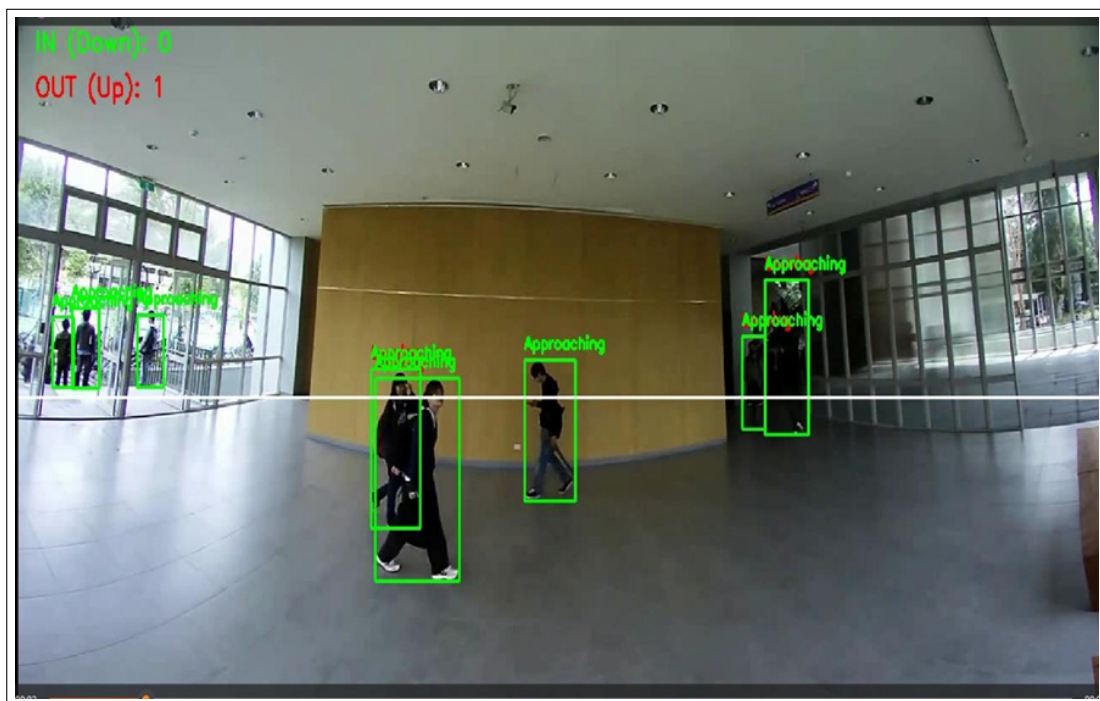


Figure 4. IN/OUT people counting output

Figure 4 presents the outcome of the directional count along with the detected and tracked pedestrians. This counting task is done only if the tracked trajectory fulfills the line crossing condition, thus making it possible to distinguish between the continuous motion in the scene and the actual entering or exiting of the scene. The IN/OUT values thus obtained become the directional count output from the tracking process.

5. Conclusion and Future Work

The proposed framework is a YOLOv9-driven system used for real-time people counting, by combining the process of human detection, Non-Maximum Suppression, centroid-based tracking, and virtual line crossing detection. The system takes both pre-recorded video clips and the webcam stream as input and tracks people and determines their movements in IN/OUT counts. The combination of detection with trajectory analysis is one of the methods to conduct automatic people monitoring in high-density areas. The CrowdHuman dataset is used to test the system under difficult conditions in terms of human density and occlusions. It will be useful for future development to make the system robust against severe occlusions, camera movement, and densely populated areas. Possible improvements of the system include multiple camera synchronization, adaptive boundary of counting, preserving human identity during tracking, deployment on edge devices, and optimizing inference pipeline.

References

- [1] Pore, Satish D., and B. F. Momin. "Bidirectional People Counting System in Video Surveillance." In International Conference on Recent Trends in Electronics, Information Communication Technology (RTEICT), IEEE, 2016, 724-727.
- [2] Perng, Jau-Woei, Ting-Yen Wang, Ya-Wen Hsu, and Bing-Fei Wu. "The Design and Implementation of A Vision-Based People Counting System in Buses." In International Conference on System Science and Engineering (ICSSE), IEEE, 2016, 1-3.
- [3] Saxena, Surbhi, and Dalpat Songara. "Design of People Counting System using MATLAB." In Tenth International Conference on Contemporary Computing (IC3), IEEE, 2017, 1-3.
- [4] Wahyuni, Elvira Sukma, Rizqi Renafasih Alinra, and Hendra Setiawan. "People Counting for Indoor Monitoring." In International Conference on Computing, Engineering, and Design (ICCED), IEEE, 2017, 1-5.
- [5] Le, Manh Cuong, My-Ha Le, and Minh-Thien Duong. "Vision-Based People Counting for Attendance Monitoring System." In 5th International Conference on Green Technology and Sustainable Development (GTSD), IEEE, 2020, 349-352.
- [6] Holla, M. Raviraja, D. Suma, and M. Darshan Holla. "Optimizing Accuracy and Efficiency

- in Real-Time People Counting with Cascaded Object Detection." *International Journal of Information Technology*, 2024, 1-14.
- [7] Wang, Chien-Yao, I-Hau Yeh, and Hong-Yuan Mark Liao. "Yolov9: Learning What You Want to Learn Using Programmable Gradient Information." In *European conference on computer vision*, Cham: Springer Nature Switzerland, 2024, 1-21.
 - [8] Patwal, Akshita, Manoj Diwakar, Vikas Tripathi, and Prabhishek Singh. "Crowd Counting Analysis using Deep Learning: A Critical Review." *Procedia Computer Science*, 2023, vol. 218: 2448-2458.
 - [9] Maktoof, Mohammed Abdul Jaleel. "Crowd Counting using Yolov5 and KCF." *Periodicals of Engineering and Natural Sciences*, 2023, vol. 11, no. 2: 92-101.
 - [10] Ahamed, Ishrath, Chamith Dilshan Ranathunga, Dinuka Sandun Udayantha, Benny Kai Kiat Ng, and Chau Yuen. "Real-Time AI-Driven People Tracking and Counting using Overhead Cameras." In *TENCON, IEEE*, 2024, 952-955.
 - [11] S. Shao, Z. Zhao, B. Li, T. Xiao, G. Yu, X. Zhang, and J. Sun, "CrowdHuman: A Benchmark for Detecting Human in a Crowd," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021. Available: <https://www.crowdhuman.org/>