IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Human Detection And People Counting Using Python

Devendrakumar R N
Department of Computer Science and Engineering
Sri Ramakrishna Institute of Technology
Coimbatore, India

Santhosh T
Department of Computer Science and Engineering
Sri Ramakrishna Institute of Technology
Coimbatore, India

Sachin Babu S
Department of Computer Science and Engineering
Sri Ramakrishna Institute of Technology
Coimbatore, India

Abdul Kalaam Aazad M
Department of Computer Science and Engineering
Sri Ramakrishna Institute of Technology
Coimbatore, India

***Abstract*—A real-time human detection and people counting system designed for surveillance, crowd monitoring, and behavioral analytics. The system leverages the YOLOv8 deep learning model for accurate and high-speed object detection, combined with the DeepSORT tracking algorithm to ensure consistent identity tracking across video frames, even under occlusion or dense crowd conditions. A motion detection module based on frame differencing and contour analysis dynamically triggers detection only when significant movement is detected, reducing unnecessary computation and improving responsiveness. The graphical user interface (GUI), developed using Python's Tkinter library, allows users to manage video streams, adjust sensitivity, and view live statistics including real-time person count and motion alerts. Experimental results show that the system achieves high detection precision (up to 92.1%) and recall (98%) with a mean average precision (mAP) of 0.971, and maintains a real-time frame rate of 20–30 FPS on GPU hardware. Comparative analysis with the SSD (Single Shot Detector) model demonstrates that YOLOv8 provides superior performance in accuracy and tracking robustness, making it well-suited for both indoor and outdoor deployments. This work demonstrates the viability of deploying intelligent vision-based systems in scalable real-world applications.**

Keywords: Computer Vision ,DeepSORT, GUI, Human Detection,kalman filter, Motion Detection, Python, Real-Time Tracking, Surveillance, YOLOv8 .

I. INTRODUCTION

1.1 OVERVIEW

In the recent past, there has been a huge increase in the requirement of effective and precise human detection and monitoring systems due to the growing demands for surveillance, multitude analysis and retail analysis. [1]Conventional systems that involve human observation or basic movement sensors have scalability and problems of real -time response capacity. To address such problems, this project suggests a real -time human detection and monitoring system through the application of deep learning techniques and computer vision.

[1]The system uses the YOLOV8 object detection model to achieve the speed -based detection and accuracy of people and the DeepSORT algorithm for long -term monitoring in video frames. These technologies provide a solid monitoring of several people even in dynamic and busy environments. To improve computational efficiency and direct resources to significant activity, a movement detection module is implemented, which begins detection only after observing a significant movement in the scene.

In addition, the GUI is made to be simple in a way that users can handle video sources, adjust the detection parameters and monitor real -time statistics. The system is made to be flexible and accessible, and admits live and stored video inputs.

This combined strategy offers a scalable and realistic solution to practical applications such as security surveillance, public safety and consumer behavior analysis.

1.2 PROBLEM DECLARATION

The human movement in urban environments full of people must be tracked efficiently in a real time. [6]The oldest detection or human monitoring is an important component of traditional surveillance systems, resulting in slow reactions, a high rate of false positives and limited adaptability in complex or congested environments. [5]These restrictions compromise the control of the crowd, the security and effectiveness of retail analysis.

In addition, most existing systems are computationally expensive and cannot track in real time, which makes them inappropriate for generalized use. The deployment is only feasible for technical stakeholders due to their hostile interfaces.

Therefore, the objective is to create an accessible, receptive and light system so that it can count and track people into a variety of real world situations.

1.3 PROJECT SCOPE

The objective of this project is to use computer vision and deep learning to create a real -time human detection and monitoring system. Combine Yolov8 for rapid detection of objects and a deep scarce for reliable monitoring of multiple objects, which allows the system to identify and follow people through frames, even in dynamic or full of people. [7]Movement detection is incorporated to improve response capacity, allowing the system to concentrate on regions with activity and reduce meaningless processing. This feature increases system detection speed and suitability for real -time applications. In addition, a intuitive graphic user interface (GUI) is created to facilitate live monitoring, movement sensitivity and simple control over video broadcasts. Users with or without technical experience can use the GUI, since it shows real -time statistics such as movement alerts and the number of people observed. The system is tested both with public data sets and with original film, and handles food for live camera and recorded videos. It is intended to be adaptable and useful in two or three different fields, such as retail analysis, monitoring of crowds and security. A work software program and thorough documentation that covers the design, implementation and performance analysis are among the final deliverables.

1.4 OBJECTIVES

The main objectives of this project are:
 To develop a real-time human detection system using YOLOv8 for fast and accurate identification of individuals in various environments. [1]To integrate DeepSORT for reliable multi-object tracking, ensuring consistent identification across frames, even in crowded or dynamic scenes. To implement a motion detection module that highlights active regions in the video feed, improving responsiveness and reducing processing load. To design and develop a user-friendly graphical user interface (GUI) for managing video sources, configuring motion detection, and displaying realtime analytics. To test and evaluate system performance using metrics such as precision, recall, mAP, and tracking accuracy under different environmental conditions. To demonstrate the practical applications of the system in areas such as security surveillance, retail behavior analysis, and public crowd monitoring. To create detailed documentation covering system design, implementation, usage, and performance results to support future improvements and real-world deployment.

II. LITERATURE REVIEW

[1]Justin Jayaraj K et al., presents a novel surveillance system that enhances person detection and tracking by integrating an improved YOLOv8 model with the DeepSORT tracking algorithm. The proposed architecture employs a DenseNet backbone to improve feature extraction efficiency, along with Feature Pyramid Networks (FPN) and Spatial Pyramid Pooling (SPP) for effective multi-scale detection. The system utilizes an anchor-free detection approach to better handle variations in object shape and scale. A curated dataset from the Google Open Images v6+ dataset, specifically focused on the "person" class, is used for training, supported by preprocessing techniques such as resizing, augmentation, and normalization. DeepSORT is used to maintain consistent identity tracking across video frames using Kalman filtering and the Hungarian for data association. The model achieves a mean Average Precision (mAP) of 0.971 at 0.5 Intersection over Union (IoU), outperforming YOLOv3 (mAP 0.88) and YOLOv5 (mAP 0.90). The system demonstrates precision, recall, and F1-scores above 0.9, indicating high detection reliability. While the model exhibits strong accuracy and robustness for real-time surveillance applications, the paper does not extensively discuss performance limitations on edge devices or resource-constrained environments. Still, the suggested architecture offers a major contribution toward intelligent surveillance systems and lays a strong basis for next developments in real-time person detection and tracking.

Singh et al. [2] suggested a real-time population counting system that combines a centroid tracking algorithm with a Single Shot Detector (SSD) built on MobileNet. The system is designed for overhead camera views and aims to count individuals in public areas such as malls and supermarkets with high accuracy and low computational cost. MobileNet is appropriate for deployment on resource-limited devices since its use as the base network instead of the heavier VGG16 greatly lowers model size and speeds inference.

The detection is performed using SSD, which localizes and classifies objects in a single forward pass. For tracking, the system employs a centroid tracking method that calculates the center of each detected bounding box and assigns a unique ID to every individual, enabling consistent tracking across frames.

The model was trained and tested on the INRIA person dataset and achieved a true positive rate (TPR) of 95.03%, false positive rate (FPR) of 0.08%, and an accuracy of 96.64%. These results demonstrate the model’s robustness, although some challenges remain under high crowd density, occlusion, and shadow interference.

Compared to the YOLO algorithm, the proposed system shows better recall and localization accuracy but has limitations in detecting smaller objects due to SSD's shallower architecture. The authors suggest that future work could involve training with more complex datasets and exploring alternative algorithms to improve performance in densely populated environments.

[3]Human detection and counting have significantly advanced, transitioning from traditional image processing techniques to deep learning-based approaches. Early methods, such as part-based detection by Mohan et al., focused on identifying individual body parts, while the Histogram of Oriented Gradients (HOG) by Dalal and Triggs became popular for pedestrian detection using edge orientation. However, these techniques often failed under challenging conditions like poor lighting, crowded scenes, and diverse human poses. A real-world failure in China, where a person on a billboard was falsely flagged as a jaywalker, highlighted the limitations of such systems. With the rise of Convolutional Neural Networks (CNNs), detection accuracy and robustness have greatly improved. [2]Models like the Single Shot Detector (SSD) process images in a single pass, enabling fast and efficient real-time object detection. The base paper builds upon this by using SSD with CNNs to detect and count people in live video streams, sending alerts when occupancy exceeds a threshold. This approach is especially useful for ensuring public safety and managing social distancing in crowded environments.

III. METHODOLOGY

The proposed system enables real-time human detection and people counting by integrating motion detection, object recognition, tracking, and a user-friendly graphical interface. Video input is captured from either live camera feeds or pre-recorded sources, each processed in a separate thread for smooth, parallel execution. Each frame is resized and preprocessed—converted to grayscale and blurred to reduce noise—before motion detection is performed using frame differencing, thresholding, and dilation techniques. Only frames with significant motion are passed for further analysis, improving processing efficiency. [1]YOLOv8 is then applied to detect human figures within the motion regions. Detected individuals are assigned unique IDs through the DeepSORT tracking algorithm, which combines motion prediction with appearance-based feature matching for consistent identity tracking. The processed results, including bounding boxes, motion alerts, and person counts, are displayed via a custom GUI. This interface supports multi-source monitoring, real-time updates, and user configuration, making the system practical and accessible for real-world deployment.

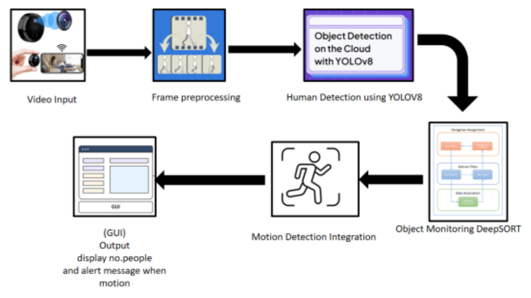

Video Input
Frame preprocessing
Object Detection on the Cloud with YOLOv8
Human Detection using YOLOV8
Object Monitoring DeepSORT
Motion Detection Integration
(GUI)
Output
display no.people
and alert message when
motion

Fig 3.1 : Overall Design

3.1 VIDEO INPUT

The system is able to process live camera food and pre -registered video sources. Each input flow is assigned to an independent thread to admit concurrent processing. This approach facilitates the gentle functioning of multiple entries without compromising system performance.

3.2 FRAME PREPROCESSING

Each video frame undergoes a structured preprocessing pipeline prior to human detection and tracking to enhance image quality and support efficient motion analysis. Initially, all input frames are resized to a fixed resolution of 800×600 pixels, ensuring uniformity across diverse video sources and reducing computational overhead. Following this, the frames are converted from color (BGR) to grayscale, as motion analysis relies on changes in pixel intensity rather than color information.To further improve motion detection accuracy, a Gaussian blur with a kernel size of (21×21) is applied, which minimizes high-frequency noise and reduces the likelihood of false positives. This smoothing operation stabilizes the pixel values across neighboring regions, thereby improving the reliability of the subsequent frame differencing step. Motion detection is performed by computing the absolute difference between the current and previous grayscale frames, producing a differential image that highlights regions with significant motion. This differential image is then processed using binary thresholding, converting it into a binary format where white pixels indicate motion. A dilation filter is applied to strengthen the motion regions by closing small gaps and merging nearby blobs. Contours are then extracted to outline the boundaries of these regions. To eliminate noise and irrelevant motion, only contours that exceed a predefined area threshold are considered valid. This selective filtering approach ensures that the detection module is triggered only in the presence of meaningful movement, thereby conserving computational resources and enhancing real-time performance.
.

3.3 HUMAN DETECTION USING YOLOV8

YOLOv8 (You Only Look Once, version 8) is a rapid but very efficient deep-learning-based object detection model designed for real-time applications. It is an even better version than the previous ones in terms of accuracy, speed, and architectural flexibility. The detection process undergoes a formalized process as described below:

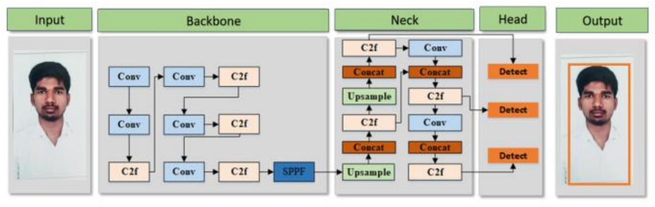


Fig 3.2 YOLOV8 Architecture

Input Processing

The input is a video frame .The frame is resized to the target dimensions (e.g., 640×640 pixels) for normal processing. Pixel values are normalized to enhance model stability.

Feature Extraction (Backbone)

YOLOv8 uses a CNN-based backbone to extract meaningful features from the image. These features include edge detection, textures, and patterns that help distinguish objects. Extracted feature maps are passed to the next layers for further processing.

Multi-Scale Object Detection (Neck)

The Feature Pyramid Network (FPN) is used to find objects at different scales. It aids in the efficient detection of small and big objects, improving precision in various conditions.

Bounding Box Prediction (Head)

YOLOv8 predicts bounding boxes for detected objects Each bounding box is assigned a class label (such as "person") and a confidence score as the probability of correct detection. The model uses anchor boxes to improve localization precision.

Non-Maximum Suppression (NMS)

Detection for the same object can overlap more than once. NMS removes duplicate bounding boxes and retains only the most confident one. This operation gives clean and correct output.

Final Output

YOLOv8 gives:

Bounding boxes around object detected.

Class labels for each detected object.

3.4 OBJECT MONITORING WITH DeepSORT

To achieve high-performance real-time tracking in videos, we utilize the DeepSORT algorithm—an advanced version of the original SORT (Simple Online and Realtime Tracking). While SORT relies solely on motion predictions for tracking, DeepSORT introduces deep learning-based appearance descriptors to improve object association over time, especially when dealing with occlusion, overlapping paths, and similar-looking objects.

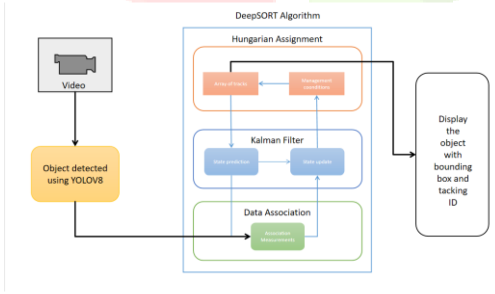


Fig 3.3 DeepSORT Algorithm

3.4.1 Overview:

DeepSORT (Deep Simple Online and Realtime Tracking) is employed to track multiple individuals across video frames by utilizing both motion information and visual appearance cues. It works in tandem with the YOLOv8 object detector, which identifies objects of interest in each frame. Using YOLO's output as input, DeepSORT assigns a unique ID to each detection and maintains consistent identity tracking across subsequent frames, even under occlusion or overlapping paths.

3.4.2 Appearance Feature Extraction:

For each person detected, [1]DeepSORT extracts the corresponding image patch and passes it through a pre-trained convolutional neural network. This network generates a compact 128-dimensional feature vector that captures the visual characteristics of the object. These descriptors help differentiate between similar-looking individuals, which is essential when multiple people follow similar motion trajectories.

3.4.3 Motion Estimation Using Kalman Filter:

To predict the next position of each tracked object, DeepSORT utilizes a Kalman filter. The filter models each track using an 8-dimensional state vector, comprising center position, height, aspect ratio, and their respective velocities. This enables smooth and accurate trajectory estimation.

3.4.4 Object Association:

Detection-to-track assignment is handled in two stages. First, motion-based matching is performed using Mahalanobis distance, comparing predicted track positions with new detections. For detections that remain unmatched, a second stage uses cosine distance between appearance vectors to find the best match. This dual strategy ensures robust tracking performance even in complex environments.

3.4.5 Hungarian Assignment:

The two cost matrices—motion and appearance—are merged into a single cost function. The Hungarian algorithm is then applied to solve the assignment problem optimally, ensuring the most accurate match between new detections and existing tracks.

3.4.6 Track Management:

To manage active and inactive tracks, DeepSORT uses a three-state lifecycle model: Tentative, Confirmed, and Deleted. A track becomes "Confirmed" only if matched in consecutive frames, while unmatched tracks are eventually removed. This management strategy prevents false tracking and helps maintain system efficiency.

3.5 MOVEMENT DETECTION INTEGRATION

To enhance system efficiency, a frame-differencing-based motion detection technique was integrated. Each video frame is first converted to grayscale and blurred using a Gaussian filter to reduce noise. The system then computes the absolute difference between the current and previous frames to detect significant pixel changes. A binary threshold is applied to the resulting difference image, followed by dilation to connect fragmented regions. Contours are extracted to localize areas of movement, and only regions exceeding a predefined area threshold are considered valid motion. This method selectively triggers human detection within dynamic regions, optimizing performance and reducing computational overhead.

3.6 GRAPHICAL USER INTERFACE (GUI)

To improve usability and accessibility, the system presents a personalized user (GUI) user interface developed with the Tkinter library in Python. The GUI is designed to accommodate users with different levels of technical experience, offering a clean design and intuitive controls. Users can add and delete video input sources, including live camera foods and pre -recorded files. Each video transmission is presented in its own dedicated tab within the interface, allowing the monitoring of multiple sources simultaneously without overlapping or confusion. This multiple structure admits real -time updates and soft transitions between sources. The GUI also provides control elements to configure the movement detection parameters. Users can enable or disable the movement detection module and adjust their sensitivity using interactive sliding controls. These configurations allow to adjust based on specific environmental conditions or application requirements. In addition to video display, the interface shows statistical data in real time, such as the total number of individuals detected and alerts triggered by movement events. When a significant activity is detected, the GUI issues visual notifications and updates the detection record instantly. This combination of control, visualization and feedback makes the system highly practical for real -time surveillance, crowd monitoring and applications to count people.

3.7 DATASET

The Common Objects in Context (COCO) dataset is used to pretreat the YOLOv8 object identification model used in this investigation. [4] With more than 330,000 photos and more than 1.5 million object instances in 80 categories, COCO is a sizable benchmark dataset intended for object detection, segmentation, and labeling tasks. [4] This dataset's "person" class, which provides a high-quality and varied representation of human figures in a variety of settings and stances, was specifically used in our study for human detection. Because of its diversity and large number of annotations, it is especially well-suited for developing reliable detection models for multi-camera and real-time surveillance systems.

IV. RESULTS AND DISCUSSION

This section assesses the real-time effectiveness of the human detection and tracking system by utilizing key metrics such as detection confidence, accuracy, and recall. To analyze the real-time performance of YOLOv8 and SSD for human detection tasks, multiple evaluation metrics were plotted and assessed over sequential video frames. These results provide insights into the accuracy, reliability, and responsiveness of each detection model.

4.1. Confidence Score Over Time

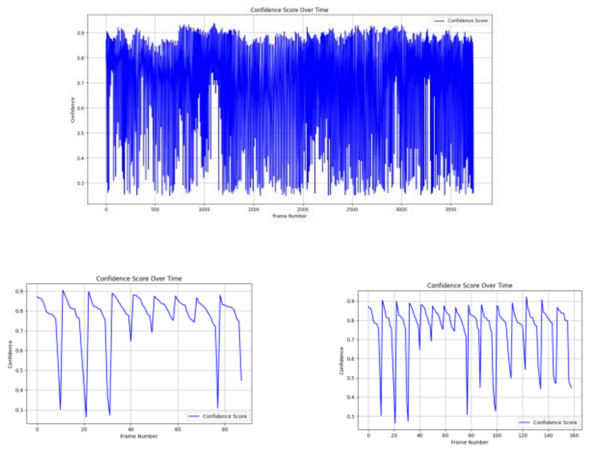


Fig 4.1 the variation of confidence scores per frame for YOLOv8

Fig 4.1 shows confidence scores across a long video sequence with over 3500 frames. Despite minor fluctuations, the majority of detection scores remained consistently above 0.8, indicating reliable performance in extended runs. Focus on shorter frame windows (first 90 and 160 frames, respectively). These demonstrate a steady trend where YOLOv8 maintains high detection confidence (~0.85–0.9), even when facing intermittent drops due to momentary occlusions or complex backgrounds. This reflects strong robustness and quick recovery.

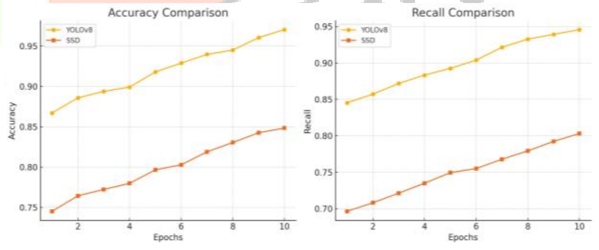
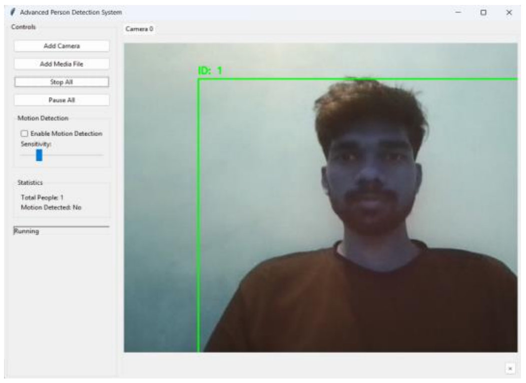


Fig 4.2. Accuracy and Recall Comparison Between YOLOv8 and SSD

Figure 4.2 provides a side-by-side comparison of accuracy and recall over 10 training epochs for both YOLOv8 and SSD: YOLOv8 shows a consistent upward trend, reaching an accuracy of ~96.5% and recall of ~94.8% by epoch 10. This demonstrates its capability to generalize well across diverse conditions. In contrast, SSD follows a slower improvement trajectory, capping around 85% accuracy and 80% recall. While it performs efficiently on lighter tasks and edge devices, it struggles to match YOLOv8's precision in dense or complex environments.

Fig 4.3 Output display the number people detected



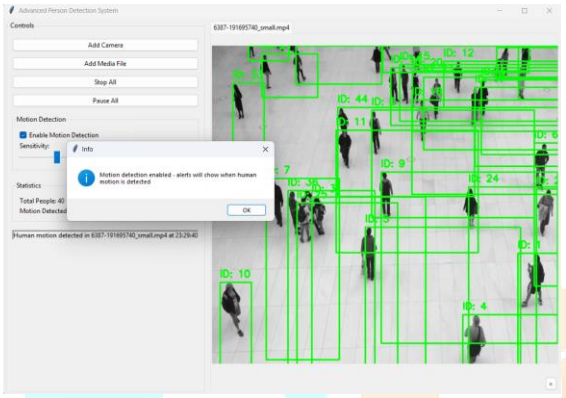


Fig 4.4 When the motion is detected on the video it get alert message

The comparative evaluation of YOLOv8 and SSD reveals significant differences in architecture, detection capability, and deployment suitability. YOLOv8 incorporates a custom backbone with Feature Pyramid Networks (FPN), enabling efficient multi-scale object detection with a real-time performance of 20–30 FPS on a GTX 1650 GPU. It achieves high detection accuracy, reaching 92.1% in indoor environments and 88.7% outdoors, along with a recall rate of 98% and a mAP@0.5 of 0.971. SSD, using MobileNet or VGG backbones, shows slightly higher speeds (30–40 FPS) on lightweight devices and achieves 96.64% precision and 95.03% recall on the INRIA dataset. However, SSD lacks built-in support for tracking or motion detection. In contrast, YOLOv8 integrates seamlessly with DeepSORT for identity tracking and features a motion-based trigger that optimizes resource usage by processing only dynamic frames. Additionally, the system includes a user-friendly Tkinter-based GUI that supports real-time video monitoring, motion alerts, and people counting across multiple sources. While SSD is ideal for deployment on resource-constrained devices, YOLOv8 proves more effective for scalable, real-time surveillance applications requiring accuracy, responsiveness, and multi-source support. The system's intuitive design and alert mechanism make it accessible for non-technical users, confirming its applicability in public safety, crowd analytics, and retail environments.

V. CONCLUSION AND FUTURE WORKS

A real-time human recognition and tracking system that combines DeepSORT for efficient multi-object tracking and YOLOv8 for high-precision object detection was reported in this work. The inclusion of a motion detection module reduced computational overhead by focusing processing on regions with activity. The system's intuitive GUI, developed with Python's Tkinter, enabled seamless monitoring of multiple video sources and live statistical feedback, making it user-friendly for both technical and non-technical users. Experimental results showed that the system consistently achieved high accuracy and real-time performance, proving its effectiveness in both indoor and outdoor environments. The comparative study further confirmed that YOLOv8 outperformed SSD in tracking consistency, detection accuracy, and integration capabilities.

Future enhancements will focus on improving scalability and deployment flexibility. These include optimizing the system for low-power edge devices such as Raspberry Pi or Jetson Nano, integrating person re-identification for long-term tracking across cameras, and enabling cloud-based monitoring and analytics. Additional features like behavioral analysis, fall detection, and crowd heatmaps are also planned to extend the system's utility in healthcare, public safety, and retail sectors.

REFERENCES

[1] J. J. K, P. V, B. Vadivel, G. G, R. K B and K. K, "Person Detection using Proposed YOLOv8 algorithm," 2025 IEEE 14th International Conference on Communication Systems and Network Technologies (CSNT), Bhopal, India, 2025, pp. 567-572, doi: 10.1109/CSNT64827.2025.10967648

[2] A. K. Singh, D. Singh, and M. Goyal, "People Counting System Using Python," Proc. 5th Int. Conf. on Computing Methodologies and Communication (ICCMC 2021), pp. 1750–1754, IEEE, 2021.

[3] R. Preethi, G. V. Praneeth, and E. S. S. Kumar, "Human Detection and Counting Using Python," International Research Journal of Engineering and Technology (IRJET), vol. 8, no. 4, pp. 780–783, Apr. 2021.

[4] T.-Y. Lin, M. Maire, S. Belongie, et al., "Microsoft COCO: Common Objects in Context," in *European Conference on Computer Vision (ECCV)*, Zurich, Switzerland, 2014, pp. 740–755.

[5] Y.-L. Hou and G. K. H. Pang, "People counting and human detection in a challenging situation," IEEE Trans. Syst., Man, Cybern. A, Syst. Humans, vol. 41, no. 1, pp. 24–33, Jan. 2011.

[6] R. Prabha, S. Mathur, B. Subramanian, S. Jain, and K. Choudhary, "Human Detector and Counter Using Raspberry Pi Microcontroller," in Proc. Int. Conf. Innovations Power Adv. Comput. Technol. (i-PACT), 2017, pp. 1–6.

[7] X. Zhao, E. Dellandréa, and L. Chen, "A People Counting System Based on Face Detection and Tracking in a Video," in Proc. IEEE Int. Conf. on Image Processing (ICIP), 2009, pp. 1–6.

[8] F. Xavier Gaya-Morey, Cristina Manresa-Yee, and Jose M. Buades-Rubio, "Deep Learning for Computer Vision-Based Activity Recognition and Fall Detection of the Elderly: A Systematic Review," arXiv preprint arXiv:2401.11790, Jan. 2024.

[9] Ekram Alam, Abu Sufian, Paramartha Dutta, Marco Leo, "Vision-based Human Fall Detection Systems using Deep Learning: A Review," arXiv preprint arXiv:2207.10952, Jul. 2022.

[10] Sun, S., Akhtar, N., Song, H., Zhang, C., Li, J., & Mian, A. (2019). Benchmark Data and Method for Real-Time People Counting in Cluttered Scenes Using Depth Sensors. IEEE Transactions on Intelligent Transportation Systems.

[11] Ahmad, M., Ahmed, I., & Adnan, A. (2019). Overhead View Person Detection Using YOLO. 2019 IEEE 10th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON).

[12] Cho, S. I., & Kang, S.-J. (2018). Real-time People Counting System for Customer Movement Analysis. IEEE Access..