International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 07 | July - 2025 SJIF Rating: 8.586 ISSN: 2582-3930

Human Density for Any Function&Violence Detection

Mr.Sanjay S [1], Mrs. N. Anandhapriya [2]

[1] *II MCA, Department of Computer Applications, Nehru Institute of Information Technology &Management, Coimbatore, Tamilnadu, India.*

[2] *Assistant Professor (SG), Department of Computer Applications, Nehru Institute of Information Technology &Management, Coimbatore, Tamilnadu, India*

---***---

Abstract - Human detection and counting in visual surveillance systems is a critical task for enhancing security, monitoring crowd behavior, and improving safety in various environments such as public spaces, retail stores, transportation hubs, and industrial settings. This paper presents a robust approach for detecting and counting individuals in real-time using computer vision and machine learning techniques. The proposed system aims to accurately identify and track human figures within video footage, providing reliable data on foot traffic, crowd density, and movement patterns. The system employs a combination of deep learning-based object detection models, such as Convolutional Neural Networks (CNNs), and more advanced architectures like You Only Look Once (YOLO) or Faster R-CNN, to detect human figures in a wide range of environments and lighting conditions. These models are trained to recognize human bodies, even in crowded or occluded settings, and to differentiate between humans and other objects in the scene. The counting functionality is achieved by tracking individual detections across video frames, ensuring that each person is counted once, even as they move through complex environments. Real-time human detection is achieved through a pipeline that processes video frames from surveillance cameras. The system provides outputs such as the number of people present in a given area, their movement trajectories, and density estimation. Alerts can be generated for abnormal crowd conditions, such as overcrowding or unusual movement patterns, which are valuable for security personnel or operational monitoring. Additionally, the system supports integration with existing surveillance infrastructure, providing a seamless solution for automated crowd management.

Key Words: Violence detection, Yolo, Video analytics, CNN, computer vision

1.INTRODUCTION

The Crowd management is a critical aspect of ensuring public safety at events, public spaces, transportation hubs, and other venues where large numbers of people gather. The ability to monitor and control crowd density can prevent overcrowding, ensure proper evacuation in emergencies, and optimize space usage. Traditional methods of crowd management often rely on manual counting or static sensors, which are not only time-consuming but also prone to inaccuracies and limitations. With the advancements in computer vision and machine learning, person counting and density estimation have become increasingly effective tools in crowd management. Computer vision systems equipped with cameras and AI algorithms can detect, track, and count individuals in real-time, even in large, dynamic environments. These systems use techniques such as object detection, tracking, and segmentation to analyze video feeds, estimate crowd density, and provide actionable insights for authorities and event organizers. The challenge lies in the accuracy and efficiency of these systems, particularly in complex scenarios such as crowded events, occlusions (where people block one another), and varying environmental conditions. Advanced algorithms, including deep learning models, have significantly improved the precision of crowd counting and density estimation, making it a valuable tool for safety, security, and urban planning.

Person counting and density estimation are critical tasks in various domains such as crowd management, security, and smart building systems. These tasks involve accurately counting the number of people in a given area and estimating crowd density, which is crucial for preventing overcrowding, ensuring safety, and optimizing the use of space. Advances in computer vision, deep learning, and sensor technologies have significantly improved the accuracy of these systems. However, challenges still persist due to occlusions, varying lighting conditions, and the complexity of different crowd behaviors. This paper explores the current methods, limitations, and potential improvements in person counting and density estimation, providing a comprehensive analysis of the critical aspects of these systems.

2.RELATED WORK:

Human Detection & Counting:
Traditional human detection systems used handcrafted features like Histogram of Oriented Gradients (HOG) and Haar cascades; however, these approaches struggled in complex environments with occlusion or scale variation. The introduction of deep learning models such as YOLO (You Only Look Once), SSD (Single Shot MultiBox Detector), and Faster R-CNN revolutionized real-time human detection with high accuracy and speed. These models have been effectively applied in crowd counting tasks, where YOLOv5 and Faster R-CNN are commonly used for their balance of speed and precision.

Crowd counting techniques are typically classified into detection-based and regression-based approaches. While detection-based models focus on identifying individual persons in a scene, regression-based models estimate crowd density maps from image features. Models like MCNN (Multi-Column CNN) and CSRNet have been successful in generating density maps, particularly in extremely dense scenes where detection models may fail due to overlapping individuals.

Violence detection has traditionally relied on motion-based features like optical flow, background subtraction, or spatiotemporal interest points (STIP). While effective in controlled environments, these methods often fail in real-world settings with noise, camera motion, or complex human interactions.Recent works incorporate deep learning architectures such as CNNs combined with Recurrent Neural Networks (RNNs) or LSTMs to learn both spatial and temporal patterns. Datasets like the Hockey Fight Dataset, Movies Fight Dataset, and RWF-2000 have been used to train such models. These deep learning methods can recognize aggressive movements, sudden gestures, and chaotic group behavior that indicate violence.

3. OVERVIEW OF THE PROJECT:

Human detection and counting in visual surveillance systems is a vital component of modern security and crowd management solutions. With growing concerns over public safety, efficient space utilization, and event monitoring, there is a critical need for intelligent systems that can accurately detect, track, and count individuals in real time. This project aims to develop a robust, AI-powered system that performs real-time human detection and counting using video feeds from surveillance cameras. The system is designed to work reliably in a variety of environments—including crowded, low-light, or occluded settings—and can support applications in security, public safety, urban planning, and smart infrastructure.

Workflow Summary:

- Live video stream is continuously captured from surveillance cameras.
- Each video frame is analyzed using YOLOv5 or Faster R-CNN for human detection.
- Detected individuals are tracked across frames using SORT or Deep SORT algorithms.
- Human counts are updated in real time, and crowd density is calculated.
- Violence detection module analyzes frame sequences using CNN-LSTM or 3D CNN models to detect aggressive or abnormal behavior.
- If overcrowding or violence is detected, alerts are automatically generated and sent to authorized personnel.
- All events and metrics are logged for future review and analysis.

Technologies and Tools Used:

- YOLOv5 / Faster R-CNN: For accurate and real-time human detection.
- SORT / Deep SORT: For robust multi-object tracking and individual identification.
- CNN-LSTM / 3D CNN: For analyzing spatiotemporal features to detect violence.
- OpenCV: For video processing, frame extraction, and visual annotations.
- Python (TensorFlow/PyTorch): For implementing and training deep learning models.
- Flask / Django: For backend integration, API management, and dashboard services.
- MQTT / WebSocket's: For real-time alert notification and communication.
- COCO / CrowdHuman Dataset: For model training and evaluation on human detection.
- XAMPP / MySQL: For storing alert logs and user interaction data.

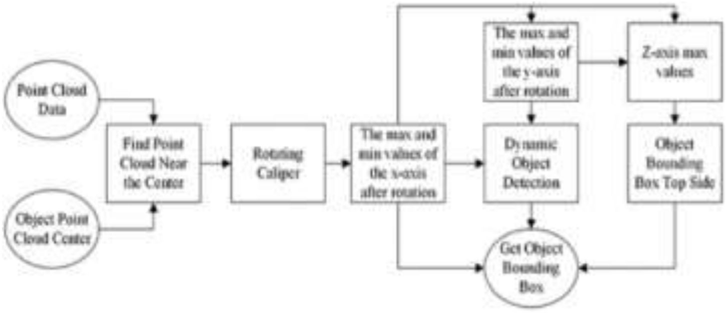

Fig 1: System Design

3.1 Module Description:

1.Data Collection Module:

This module is responsible for acquiring real-time video input from surveillance cameras installed in the monitoring environment. It serves as the entry point for the entire system, continuously streaming footage for analysis. Additionally, it can optionally collect metadata such as the geographic location, camera ID, and orientation, which can enhance context-aware analysis and reporting.

2. Frame Extraction and Preprocessing Module:

Once the video stream is acquired, this module extracts individual frames at regular intervals to enable frame-by-frame analysis. To optimize performance and ensure compatibility with deep learning models, the frames are preprocessed through operations such as resizing, normalization, and color adjustment. These steps prepare the data for efficient and accurate human detection.

3.Human Detection Module:

This is the core module that applies advanced deep learning-based object detection models—such as YOLO or Faster R-CNN—to identify human figures within each video frame. The models are trained to detect human bodies in a variety of poses and environmental conditions. Upon detection, the module marks each individual with a bounding box, distinguishing them from other objects or background elements.

4.Human Tracking Module:

After detection, this module tracks each identified human across consecutive video frames to maintain continuity. Using tracking algorithms like Kalman Filter or SORT (Simple Online and Realtime Tracking), it assigns a unique ID to each person. This

ID ensures that individuals are not counted multiple times, even as they move across the camera's field of view.

5.Counting & Event Logging Module:

This module is responsible for maintaining an accurate count of individuals detected and tracked by the system. It ensures that each unique person is only counted once and logs key events, such as entry or exit, along with timestamps. Movement trajectories and crowd flow data can also be logged to support deeper analysis of behavioral patterns and density.

6.Alert & Notification Module:

An optional yet critical component, this module monitors crowd density in real time and triggers alerts when predefined thresholds are exceeded. These alerts can be sent to security personnel or control systems, enabling proactive management of overcrowding, potential emergencies, or abnormal movement patterns. This functionality enhances situational awareness and supports rapid response actions.

4. Methodology:

The proposed system integrates deep learning-based human detection and violence recognition models into a real-time surveillance pipeline to monitor human density and identify violent behaviors in public or private spaces. The methodology is structured into the following stages:

4.1 Data Acquisition

Live video streams are captured using surveillance cameras installed at strategic locations such as event venues, public places, or indoor facilities. Publicly available datasets such as **COCO**, **CrowdHuman**, and **RWF-2000** are used for pre-training and testing the models to ensure generalization to various environmental conditions.

4.2 Human Detection

Each frame extracted from the video stream undergoes object detection using **YOLOv5** or **Faster R-CNN** models. These models are selected for their real-time processing capability and high accuracy in detecting multiple humans in cluttered scenes.

4.3 Human Tracking & Density Estimation

To avoid duplicate counts and support continuous monitoring, detected humans are tracked using the **Deep SORT** algorithm, which leverages Kalman filters and appearance descriptors. Human count is aggregated frame-by-frame to estimate density and flag overcrowded scenarios.

4.4 Violence Detection

To detect violent or abnormal behavior in crowds, each video segment is processed using a **CNN-LSTM** or **3D CNN** model. These models learn spatiotemporal patterns associated with aggressive actions such as fights, sudden gestures, or chaotic movement. Pre-trained on datasets like **RWF-2000** and **Surveillance Fight Dataset**, the system can classify violence with a high degree of accuracy.

4.5 Alert Generation & Management

When the system detects high crowd density or violent activity beyond a set threshold, it triggers alerts. These alerts are managed through a web-based backend built on **Flask** or **Django**, and delivered in real-time using **MQTT** or **WebSockets** protocols. Notifications may include timestamps, snapshots, and location data, enabling authorities to respond swiftly.

4.6 Storage and Logging

All incident data, including video frames, event timestamps, crowd statistics, and classification results, are logged and stored in a **MySQL** database via a **XAMPP** local server setup. This supports audit trails, report generation, and further model improvement.

5.0 SYSTEM SPECIFICATION

Requirement Specification is the part of the project which gives the details about the hardware and software requirements of our project. It also details the features of the programming language used.

5.1 Hardware Requirements

Table 1: Hardware Requirement

Component	Specification
Processor	Intel
RAM	4GB
Main Memory	8GB RAM
Processing Speed	600MHZ
Hard Disk Drive	1TB
Keyboard	104 Keys
Camera	Standard HD Resolution

5.2 Software Requirement

Table 2: Software Requirement

Component	Specification
Front-end	HTML, CSS, JavaScript
Back-end	Python, PHP
Database	MySQL
Server Environment	XAMPP(Apache,MySQL.PHP,php MyAdmin)
Dataset Format	CSV
Development IDE	Anaconda
Operating System	Windows 11

International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 07 | July - 2025 SJIF Rating: 8.586 ISSN: 2582-3930

6. SYSTEM IMPLEMENTATION

System implementation is the final phase that is putting the utility into action. Implementation is the state in the project where theoretical design turned into a working system. The most crucial stage is achieving anew successful system and giving confidence in the new system that it will work efficiently and effectively. The system is implemented only after thorough checking is done and it is found working according to the specifications. System implementation is in the final phase. i.e., integrating all modules into live action. Implementation is the state in the project where theoretical design is turned into a working system. The implementation stage is a system project in its own right. It involves careful planning, design, investigation of the current system and constraints on implementation, design of methods to achieve change over, and evolution method. Once the planning has been completed the major effort is to ensure that the programs in the system are working properly. At the same time concentrate on training user staff.

The major implementation procedures are: -

- Build Application
- User Signing
- Configuring Server(Apache Tomcat)
- Load Modules using the JSON
- Finally Record/Test any web-based application

6.1 Equipment Installation

Anaconda is the free and open-source Python and R programming language distribution that is simple to set up. Anaconda is a software environment for mathematical computation, computer science, predictive analysis, and deep learning. Anaconda 5.3 is the most recent distribution, which was launched in October of 2019. It contains the module, an environmental manager, and the library at over 1000 open-source packagers, all of which come with free community support.

Anaconda navigator is the graphics users' interface (GUI) for desktop that comes with the Anaconda distribution. It helps us to use the Anaconda distribution's software and control conda packagers, environmental, and networks withheld having to use the commanded line command. It is most compatible for the system, Mac OS X, Linux.

XAMPP is a free and open-source cross-platform web server package that includes **Apache**, **MySQL/MariaDB**, **PHP**, and **Perl**. It allows developers to set up a local server environment on their own machine for testing and development purposes.

1.Purpose of Installation

The installation of XAMPP provides a ready-to-use development environment by bundling essential software components in one package. It eliminates the need for individually downloading and configuring Apache, PHP, and MySQL.

2. Pre-Installation Requirements

○ Operating System: Windows, Linux, or macOS
○ Administrator privileges to install and configure system services
○ Disk space for storing XAMPP files and web projects

3. Installation Steps (Theory)

○ **Download**: The installer package is downloaded from the official Apache Friends website.
○ **Execution**: Upon execution, the setup wizard guides the user through the installation.
○ **Component Selection**: Users select which components they want to install (e.g., Apache, MySQL, phpMyAdmin).
○ **Installation Directory**: The user chooses where the software should be installed (typically C:\xampp on Windows).
○ **Service Configuration**: Apache and MySQL can be set to run as system services (optional).
○ **Completion**: Once installed, the XAMPP Control Panel is used to manage services.

Post-Installation

○ Users can launch Apache and MySQL using the XAMPP Control Panel.
○ Web applications can be stored in the htdocs folder.
○ phpMyAdmin can be accessed through a browser via http://localhost/phpmyadmin to manage databases.

7. SAMPLE OUTPUT

Figure 2: Main Dashboard

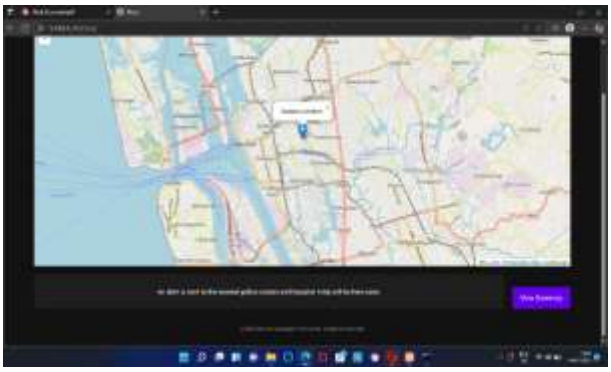

Figure 3: Live Location Tracking

Figure: 4 Admin Page

8. CONCLUSIONS

Human detection and counting in visual surveillance systems play a vital role in enhancing security, monitoring crowd behavior, and ensuring public safety across diverse environments such as public spaces, transportation hubs, retail stores, and industrial settings. The integration of advanced computer vision and machine learning techniques, particularly deep learning models like YOLO and Faster R-CNN, has significantly improved the accuracy and efficiency of identifying and tracking individuals in real-time, even in challenging conditions involving occlusions and varying lighting.

By leveraging robust detection algorithms combined with effective tracking methods, the system provides precise counting and reliable data on crowd density and movement patterns. This enables proactive crowd management by alerting authorities about overcrowding or abnormal activity, thus facilitating timely intervention and risk mitigation.

Furthermore, the modular design of the system ensures seamless integration with existing surveillance infrastructure, making it a scalable and practical solution for automated crowd monitoring. Despite ongoing challenges such as occlusions and complex crowd dynamics, continuous advancements in deep learning and sensor technologies promise further enhancements in accuracy and reliability.

Overall, the proposed approach contributes significantly to improving safety, security, and operational efficiency in crowded environments, supporting smarter urban planning and responsive crowd control measures.

ACKNOWLEDGEMENT

I hereby acknowledge that the project entitled **"Human Density for Any Function& Violence Detection Using AI"** would not have materialized with the guidance and help I received from the concerned authorities. I express my sincere gratitude to all those who have spared their contribution in this effort. The project would not be a success without the constant and valuable guidance of my Internal Project Supervisor **Mrs. N. Anandha Priya MCA,** Asst. Professor, Department of Computer Applications whose support simulating suggestions and encouragement in all sorts of help as and when I required.

Last but not the least I am grateful to my parents, friends, and also the people who supported me directly and indirectly during the project.

REFERENCES

- Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You Only Look Once: Unified, Real-Time Object Detection. CVPR, 779–788. https://doi.org/10.1109/CVPR.2016.91
- Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. NeurIPS, 91–99.
- Wojke, N., Bewley, A., & Paulus, D. (2017). Simple Online and Realtime Tracking with a Deep Association Metric. IEEE International Conference on Image Processing (ICIP).
- Dalal, N., & Triggs, B. (2005). Histograms of Oriented Gradients for Human Detection. CVPR, 886–893.
- Zhang, Y., Zhou, D., Chen, S., Gao, S., & Ma, Y. (2016). Single-Image Crowd Counting via Multi-Column Convolutional Neural Network. CVPR, 589–597.
- Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C., & Berg, A. (2016). SSD: Single Shot MultiBox Detector. ECCV, 21–37.
- Bochinski, E., Eiselein, V., & Sikora, T. (2017). High-Speed Tracking-by-Detection Without Using Image Information. ICIP.
- Xu, Y., & Corso, J. J. (2016). Evaluation of Supervoxel Methods for Early Video Processing. CVPR, 1202–1211.