

Automated Human Detection and Overcrowding Prediction Using Deep Learning and Django

Ms. Sneha Jyothi S¹, Ms. Jane Pricilla E², Mr.M Senthilkumar³, Dr.J Rajeswari⁴

¹Msc Data Science, Nehru Arts and Science College, Coimbatore, Tamilnadu, India snehajyothisudheer@gmail.com

²Msc Data Science

Nehru Arts and Science College, Coimbatore, Tamilnadu, India janejjrl@gmail.com

³Assistant Professor

Department Of Computer Science and Data Science Nehru Arts and Science College, Coimbatore, Tamilnadu, India senthilmsc09@gmail.com

⁴Assistant Professor, Department Of Computer, Science and Data Science Nehru Arts and Science College, Coimbatore, Tamilnadu, India rajeswarikrishna82@gmail.com

KEYWORDS

ABSTRACT

Effective crowd monitoring is essential for public safety, space management, and preventing overcrowding-related risks. This project introduces an Automated Human Detection and Overcrowding Prediction System using Django and deep learning-based computer vision to provide a scalable and efficient solution for real-time crowd monitoring.

The system processes both uploaded video files and live surveillance feeds, detects individuals in each frame, and generates alerts if the human count surpasses a predefined threshold, indicating potential overcrowding. To enhance usability, the system supports real-time monitoring with dynamic visualization and automated notifications via email or SMS when overcrowding is detected.

The core functionality relies on an advanced object detection model implemented in HumanDetection.py, which extracts frames from videos, identifies people, and accurately counts them. The Django-based backend manages video processing, real-time data storage, a RESTful API for seamless integration with external applications, and an intuitive web dashboard. The frontend offers interactive heatmaps, crowd density analytics, and historical trend reports to help administrators make informed decisions.

This system has broad applications in public safety, event management, workplaces, transportation hubs, and smart city initiatives. Future enhancements may include IoT-based automated crowd control, anomaly detection for unusual crowd behavior, AI-powered predictive analytics for crowd flow forecasting, and cloud-based deployment for large-scale accessibility. Additionally, integrating facial recognition (if permitted by privacy regulations) or posture analysis could improve security by identifying suspicious activities.

By leveraging deep learning, real-time analytics, and web-based automation, this project provides an intelligent, proactive, and scalable solution for monitoring and managing crowd density, ensuring safer and more efficient public spaces.

INTRODUCTION

Crowd management is a critical aspect of public safety, event coordination, and space optimization. Overcrowding in public spaces, workplaces, or transportation hubs can lead to safety hazards, inefficiencies, and regulatory violations. Traditional methods of monitoring crowd density rely on manual supervision, which is time-consuming, prone to human error, and ineffective in large-scale environments. To address these challenges, automated human detection and overcrowding prediction systems powered by artificial intelligence (AI) and computer vision have gained significant attention as they offer real-time, accurate, and scalable solutions.

This project presents an Automated Human Detection and Overcrowding Prediction System, designed to detect individuals from both uploaded video files and live surveillance feeds while determining whether the

crowd density surpasses a pre-defined threshold. The system leverages deep learning-based object detection models to analyze video frames, accurately identify people, and generate alerts when overcrowding is detected. Real-time monitoring, automated notifications (via email/SMS), and interactive analytics enhance the system's efficiency, making it a robust tool for proactive crowd management.

The system consists of four key components:

- Deep Learning-Based Human Detection – A pre-trained object detection model processes video frames, accurately identifies individuals, and counts them to determine crowd density.
- Django-Based Backend – Manages video uploads and live feed processing, handles API communication, and stores real-time and historical detection results for trend analysis.
- User Interface with Advanced Visualization – Provides a web-based platform with dynamic heatmaps, real-time overcrowding alerts, interactive dashboards, and historical crowd density analytics for decision-making.
- Automated Alert System – Sends instant notifications via email or SMS when overcrowding is detected and allows integration with IoT-based crowd control mechanisms.

This solution is particularly useful for public safety monitoring, event management, transportation hubs, workplaces, and smart city applications. Future enhancements may include AI-powered predictive analytics for crowd flow forecasting, anomaly detection for unusual crowd behavior, cloud-based deployment for large-scale access, and facial recognition-based security features (if permitted by privacy regulations). By utilizing deep learning, real-time analytics, and web-based automation, this project aims to provide an intelligent, proactive, and scalable tool for modern crowd monitoring applications.

I. OBJECTIVES

A. Develop an Advanced Human Detection and Counting System

- Implement a deep learning-based object detection model to identify and count people in video frames with high accuracy.
- Optimize detection for different lighting conditions, camera angles, occlusions, and varying crowd densities to ensure robust performance.
- Support both pre-recorded videos and real-time surveillance feeds for greater flexibility in monitoring.

B. Predict Overcrowding Scenarios and Provide Automated Alerts

- Set a predefined threshold for crowd density and trigger alerts when the limit is exceeded.
- Provide real-time notifications via email, SMS, or in-app alerts for instant action.
- Integrate visual indicators such as heatmaps and congestion markers to highlight overcrowded areas dynamically.
- Enable anomaly detection to identify unusual crowd behavior and potential security threats.

C. Build a Scalable Django-Based Web Application

- Develop a user-friendly, interactive dashboard for users to upload videos, monitor live feeds, and visualize crowd density.
- Use Django as the backend to handle video processing, real-time data streaming, and API communication.
- Provide RESTful API endpoints for seamless integration with external applications, such as smart surveillance systems and security platforms.

D. Enhance Data Storage, Management, and Analytics

- Maintain a structured database to store: Video processing history (processed files and timestamps).
Detected crowd sizes and density trends over time. Alert logs for tracking past overcrowding incidents.
- E. Enable real-time and historical trend analysis to help users study patterns and improve crowd management strategies.

Improve System Usability, Performance, and Scalability

- Optimize the system to handle large video files efficiently with advanced frame extraction and processing techniques.
 - Ensure scalable architecture that supports cloud-based deployment for large-scale accessibility.
 - Implement multi-user access control to allow different stakeholders (e.g., security personnel, event managers) to interact with the system.
- F. *Enable Future Expansion for Smart Crowd Monitoring and Control*
- Lay the groundwork for IoT-based automated crowd control mechanisms (e.g., automatic gate closures, ventilation adjustments).
 - Explore real-time heatmap-based crowd density visualization for enhanced monitoring.
 - Investigate predictive analytics for crowd flow forecasting using AI-powered data models.
 - Integrate social distancing compliance monitoring for health and safety regulations.
 - Consider facial recognition (where permitted) or behavior analysis for enhanced security applications.
- By achieving these objectives, this project provides a robust, intelligent, and automated solution for crowd monitoring and management, enhancing safety, regulatory compliance, and space utilization in various public and private settings.

II. MODULE DESCRIPTION

The Automated Human Detection and Overcrowding Prediction System consists of multiple interconnected modules that enable video processing, real-time monitoring, human detection, alert generation, and data visualization. The system efficiently handles both uploaded video files and live surveillance feeds, ensuring a robust and scalable crowd monitoring solution.

The current implementation includes the following key modules:

A. *Video Upload, Live Feed, and Preprocessing Module*

- Video Upload: Allows users to upload video files through a web-based interface for analysis.
- Live Surveillance Feed Support: Enables real-time video streaming for continuous monitoring of crowded spaces.
- Frame Extraction: Extracts frames at specific intervals from uploaded or live video feeds for efficient processing.
- Preprocessing:
 - Resizes frames and normalizes pixel values for improved model performance.
 - Enhances video frames for better detection under various lighting conditions and camera angles.

B. *Human Detection and Crowd Counting Module*

- Utilizes a deep learning-based object detection model (such as YOLO, Faster R-CNN, or OpenCV-based detection) to identify and count individuals in each frame.
- Processes extracted frames to detect human figures, overlay bounding boxes, and assign unique IDs for better tracking.
- Ensures high detection accuracy across various environments, occlusions, and crowd densities.
- Implements multi-camera fusion techniques (if multiple cameras are integrated) for enhanced monitoring.

C. *Overcrowding Prediction and Automated Alert Module*

- Threshold-Based Alerting: Compares the detected human count with a predefined overcrowding threshold.
- Automated Notifications:

- Triggers alerts via on-screen indicators, email, or SMS when overcrowding is detected.
- Sends instant alerts to security personnel or event managers for immediate action.
- Anomaly Detection: Identifies unusual crowd behavior, such as sudden crowd surges or irregular movements, using AI-powered pattern recognition.
- Historical Alert Storage: Logs previous alerts and detections for later analysis and decision-making.
- D. *Django-Based Backend and API Module*
 - Manages user requests, video processing, and real-time data storage through Django's web framework.
 - Stores processed data in a structured database, including:
 - Uploaded video files or live feed data.
 - Detected crowd sizes and timestamps.
 - Alerts and notifications triggered.
 - RESTful API Integration:
 - Provides API endpoints for seamless communication between the detection module, user interface, and external applications.
 - Allows integration with IoT-based crowd control mechanisms, such as automatic gate closures or ventilation adjustments.
- E. *Advanced Data Visualization and User Interface Module*
 - Interactive Dashboard: Displays:
 - Live and past detection results with overlays of detected individuals.
 - Dynamic heatmaps for crowd density visualization.
 - Graphical reports of crowd trends over time.
 - User Management System: Supports multi-user access with different roles (e.g., admin, security personnel, event manager).
 - Historical Data Analysis:
 - Enables users to review past detections, alerts, and crowd density trends.
 - Provides downloadable reports for compliance and regulatory documentation.

By integrating these modules, the Automated Human Detection and Overcrowding Prediction System offers a highly efficient, scalable, and proactive solution for modern crowd monitoring applications, ensuring public safety, regulatory compliance, and smart space management.

ALGORITHMS USED IN THE AUTOMATED HUMAN DETECTION AND OVERCROWDING PREDICTION SYSTEM

The Automated Human Detection and Overcrowding Prediction System employs a combination of deep learning, computer vision, and backend processing techniques to achieve accurate human detection, real-time monitoring, and alerting. Below are the key algorithms used:

A. Object Detection Algorithm (YOLO / Faster R-CNN / SSD)

- Purpose: Identifies and counts the number of people in video frames for crowd monitoring.
- Implementation:
 - Uses pre-trained deep learning models (e.g., YOLO (You Only Look Once), Faster R-CNN (Region-Based Convolutional Neural Network), or SSD (Single Shot MultiBox Detector)) to detect humans in each frame.
 - Processes real-time video feeds or pre-recorded video frames for crowd estimation.
- Why Used?
 - High accuracy in detecting human figures, even in dense crowds or occlusions.
 - Fast and efficient processing, suitable for both video frame-by-frame detection and real-time surveillance.
 - Works under different lighting conditions, camera angles, and environmental variations.

B. Frame Extraction and Preprocessing Algorithm

- Purpose: Converts video files into individual frames to optimize detection efficiency.
- Implementation:
 - Uses OpenCV to extract frames at fixed time intervals (e.g., every nth frame).
 - Preprocesses frames by resizing, normalizing pixel values, and enhancing contrast for better detection accuracy.
- Why Used?
 - Reduces computational load by skipping redundant frames while maintaining detection accuracy.
 - Ensures efficient detection by providing uniformly processed frames for the AI model.

C. Overcrowding Prediction Algorithm

- Purpose: Determines whether the detected crowd size exceeds a predefined safety threshold.
- Implementation:
 - Compares the detected human count per frame against a configurable threshold value.
 - If the count surpasses the threshold, the system triggers alerts (e.g., on-screen warnings, notifications, or automated actions).
- Why Used?
 - Provides real-time detection and alerting for overcrowding situations.
 - Enables proactive crowd control in public spaces, workplaces, and event venues.

D. Non-Maximum Suppression (NMS) Algorithm

- Purpose: Eliminates redundant detections by keeping only the most confident bounding boxes per detected person.
- Implementation:
 - Applies NMS to filter overlapping bounding boxes, keeping the one with the highest confidence score.
- Why Used?
 - Prevents duplicate counting of individuals in a crowded scene.
 - Enhances detection accuracy by reducing false positives and redundant detections.

E. Automated Alerting and IoT-Based Integration Algorithm

- Purpose: Sends alerts and triggers automated actions when overcrowding is detected.
- Implementation:
 - If human count exceeds the threshold, the system triggers email, SMS, or dashboard alerts.
 - Can integrate with IoT-based crowd control systems (e.g., automatic gate closures, ventilation adjustments, or traffic redirection).
- Why Used?
 - Enables automated response mechanisms to handle overcrowding in real time.
 - Improves public safety and regulatory compliance.

F. Django-Based Web Processing Algorithm

- Purpose: Manages data flow, user interactions, and back-end processing.
- Implementation:
 - Uses Django's Model-View-Controller (MVC) architecture to:
 - * Handle video uploads and processing requests.
 - * Store results in a structured database (e.g., SQLite, PostgreSQL, or MySQL).
 - * Provide API endpoints for frontend-backend communication.
- Why Used?
 - Offers a scalable and modular structure for handling multiple users and concurrent video processing.
 - Ensures efficient data storage and retrieval for future analysis.

ACKNOWLEDGMENT

We would like to express our sincere gratitude to everyone who contributed to the successful completion of this research project, "Automated Human Detection and Overcrowding Prediction System."

First and foremost, we extend our heartfelt appreciation to our mentors and advisors for their invaluable guidance, technical insights, and continuous support throughout this research. Their expertise and encouragement played a crucial role in shaping the direction of this work.

We would also like to acknowledge the contributions of our colleagues and peers, whose discussions, suggestions, and constructive feedback helped refine our approach and enhance the system's efficiency.

Special thanks to the developers and researchers in the field of deep learning and computer vision, whose open-source frameworks and pre-trained models significantly facilitated the implementation of this project. The contributions from the YOLO, Faster R-CNN, OpenCV, and Django communities were instrumental in developing an effective and scalable solution.

Furthermore, we are grateful to our institution/organization for providing the necessary resources, infrastructure, and research environment to carry out this project successfully. Their unwavering support enabled us to explore innovative methodologies in AI-driven crowd monitoring.

Finally, we extend our deepest appreciation to our family and friends for their constant motivation and encouragement throughout the research journey.

This work would not have been possible without the collective efforts of all those mentioned, and we are truly thankful for their support in bringing this research to fruition.

REFERENCES

- [1] Redmon, J., Farhadi, A. (2018). "YOLOv3: An Incremental Improvement." arXiv preprint arXiv:1804.02767.
 - Provides an in-depth explanation of the YOLO (You Only Look Once) object detection model, which is widely used for real-time human detection in videos.
- [2] Ren, S., He, K., Girshick, R., Sun, J. (2015). "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks." Advances in Neural Information Processing Systems (NeurIPS).
 - Describes the Faster R-CNN model, a high-accuracy object detection method used in video-based surveillance.
- [3] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., Berg, A. C. (2016). "SSD: Single Shot MultiBox Detector." European Conference on Computer Vision (ECCV).
 - Discusses the SSD (Single Shot Detector) model, an alternative to YOLO and Faster R-CNN for object detection.
- [4] Dalal, N., Triggs, B. (2005). "Histograms of Oriented Gradients for Human Detection." IEEE Conference on Computer Vision and Pattern Recognition (CVPR).
 - Presents the HOG (Histogram of Oriented Gradients) method, which is a classical approach for human detection in images and videos.
- [5] "Deep Learning for Computer Vision" by Rajalingappaa Shanmugamani.
 - Covers the implementation of deep learning techniques for object detection and image analysis.
- [6] "Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow" by Aurelien Geron.
 - Provides insights into deep learning models, including CNNs (Convolutional Neural Networks) for image and video processing.