CROWD MANAGEMENT USING AI

A PROJECT REPORT

21CSC305P –MACHINE LEARNING

(2021 Regulation)

III Year/ V Semester

Academic Year: 2024 -2025

Submitted by

MUNAGALA CHANDRA VAMSI REDDY [RA2211026010161]
AKULA SATYA SESHA SAI [RA2211026010168]
JAMI JASWANTH [RA2211026010173]
PALAMETI REDDY LAKSHMI MANOJ [RA2211026010179]

Under the Guidance of

Dr. M. Ferni Ukrit
Associate Professor
Department of Computational Intelligence

in partial fulfillment of the requirementsfor the degree of

BACHELOR OF TECHNOLOGY
in
COMPUTER SCIENCE ENGINEERING
with specialization in
ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

SCHOOL OF COMPUTING
COLLEGE OF ENGINEERING AND TECHNOLOGY
SRM INSTITUTE OF SCIENCE ANDTECHNOLOGY
KATTANKULATHUR- 603 203

NOVEMBER 2024

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY
KATTANKULATHUR – 603 203

BONAFIDE CERTIFICATE

Certified that **21CSC305P - MACHINE LEARNING** project report titled "**CROWD MANAGEMENT USING AI**" is the bonafide work of "**MUNAGALA CHANDRA VAMSI REDDY[RA2211026010161], AKULA SATYA SESHA SAI[RA2211026010168], JAMI JASWANTH[RA2211026010173], PALAMETI REDDY LAKSHMI MANOJ[RA2211026010179]**" who carried out the task of completing the project within the allotted time.

SIGNATURE
Dr. M. Ferni Ukrit
Course Faculty
Associate Professor
Department of Computational Intelligence
SRM Institute of Science and Technology
Kattankulathur

SIGNATURE
Dr. R. Annie Uthra
Head of the Department
Professor
Department of Computational Intelligence
SRM Institute of Science and Technology
Kattankulathur

ABSTRACT

This project presents an AI-powered system designed for crowd management by utilizing computer vision to assess crowd density, monitor social distancing, and count individuals in real time. Built on machine learning techniques like YOLO (You Only Look Once) and HOG (Histogram of Oriented Gradients) combined with OpenCV, this system processes live video feeds to detect people and generate density estimates. By tracking individual movements and producing heatmaps for high-traffic areas, the system allows event organizers and venue managers to respond proactively to crowding issues. The system's output includes real-time notifications via a dashboard, enhancing public safety and operational oversight in dynamic environments. This project introduces a robust AI-driven crowd management solution that integrates real-time crowd analysis, density estimation, and social distancing monitoring to improve safety and operational efficiency in public spaces. Leveraging advanced computer vision and machine learning algorithms, including YOLOv5 for object detection and HOG for human detection, the system analyses live video feeds to count individuals, estimate crowd density, and generate heatmaps of high-density areas. The platform includes a user-friendly dashboard that displays actionable insights and alerts, helping event organizers and venue managers make data-driven decisions to address crowding, ensure adherence to social distancing guidelines, and enhance visitor safety. The core functionalities involve: (1) crowd counting to determine the number of people within a space, (2) crowd density estimation to measure congestion levels over time, and (3) physical distancing checks, where proximity violations trigger automated alerts. Heatmaps overlay the accumulated crowd density data onto spatial visuals, highlighting zones with high activity levels to support dynamic crowd control and urban planning initiatives.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS

YOLO - You Only Look Once

HOG - Histogram of Oriented Gradients

SVM - Support Vector Machine

CNN - Convolutional Neural Network

CFPM – Crowd Flow Prediction and Management

OD – Object Detection

ABM – Attention Based Models

RNN – Recurrent Neural Networks

LSTM – Long Short-Term Memory

AI – Artificial Intelligence

ML – Machine Learning

MMDF – Multi Model Data Fusion

FL – Federated Learning

CV – Computer Vision

HCLM – Hybrid CNN-LSTM Model

EC – Edge Computing

CHAPTER 1

INTRODUCTION

With the increasing need for efficient crowd management in public spaces, particularly in the context of public safety and health compliance, artificial intelligence has become a critical tool for real-time monitoring and control. This project leverages AI and computer vision to provide a comprehensive solution for crowd management, offering capabilities in crowd counting, heat map generation, and physical distancing detection to help venue managers, event organizers, and city planners manage public spaces more effectively.

1.1 BACKGROUND AND MOTIVATION:

- **Ai in Crowd Management:** With the rapid growth of urban populations and the increasing complexity of public spaces, managing crowd dynamics has become a critical challenge for safety and efficiency. Crowded environments, such as stadiums, malls, airports, and transportation hubs, can become hazardous, especially when physical distancing measures are required, such as during the COVID-19 pandemic. Traditional crowd management techniques often rely on manual monitoring, which is time-consuming and prone to human error. In response, artificial intelligence (AI) and computer vision have emerged as vital tools for automating crowd monitoring and ensuring public safety in real-time. AI enables real-time crowd counting, heatmap generation, and physical distancing detection, which help improve crowd flow, reduce risks of overcrowding, and maintain public health guidelines.
- **Importance of Real-time Monitoring:** Public safety, health compliance, and effective space management are top priorities in managing large crowds, especially in high-traffic venues. As crowd sizes increase, the risk of overcrowding and associated safety hazards such as panic, congestion, and injury rise significantly. In this context, AI technologies, such as those used for crowd counting and heatmap generation, have proven to be invaluable in monitoring and controlling these challenges efficiently. The ability to provide real-time insights into crowd density, individual identification, and potential physical distancing violations enables event organizers, venue managers, and city planners to make informed decisions quickly, improving overall crowd safety.

- **Ai-Driven Crowd Management Solutions:** Artificial intelligence, particularly machine learning (ML) and computer vision models like YOLO (You Only Look Once), offers significant advantages for real-time crowd analysis. These technologies can track and analyze thousands of data points simultaneously, ensuring accurate and scalable solutions for crowd monitoring. AI-driven systems can dynamically adjust to varying conditions, detect individuals with high precision, and generate valuable insights such as density maps, crowd movement patterns, and alerts for physical distancing violations. As the demand for smarter, safer public spaces continues to grow, AI technologies are central to providing sustainable, scalable solutions to crowd management.
- **Challenges In Manual Crowd Management:** Traditional crowd management techniques are not well-suited to handle the dynamic nature of large crowds in real time. Manual observation is limited by human capacity, time constraints, and the inability to track large groups simultaneously. These systems also struggle to provide accurate, consistent data regarding crowd density or behavior, often leading to suboptimal decision-making. In contrast, AI-driven crowd management systems can process large amounts of real-time video data, continuously detecting, tracking, and analyzing crowd behaviors with high accuracy, thus addressing the limitations of manual systems and enhancing overall operational efficiency.
- **The Role of Yolov5x in Crowd Management:** In this project, the YOLOv5x model is leveraged for detecting and counting individuals within crowds. YOLOv5x is an advanced object detection model known for its speed and accuracy, making it ideal for real-time applications. By analyzing video streams, YOLOv5x detects people in individual frames and generates accurate counts and bounding boxes, which are then used for further analysis. This model also allows for the creation of heatmaps and the monitoring of physical distancing, facilitating real-time interventions when necessary. YOLOv5x represents a powerful tool in the AI-driven approach to crowd management, enabling the system to function effectively in diverse and high-traffic environments.

1.2 PROBLEM STATEMENT AND OBJECTIVES:

1. **Problem Statement:** The goal of this project is to develop an AI-powered system using the YOLOv5x model for crowd monitoring and analysis. This system processes input video to generate three types of outputs:

- An annotated video highlighting each detected individual with bounding boxes and a total crowd count.
- A heatmap visualization showing high-density areas.
- A physical distancing alert video that flags individuals not maintaining sufficient distance.
- Through these capabilities, the system aims to support efficient crowd management, enhancing public safety and adherence to health protocols in high-traffic environments.

2. **Objectives of the Study:** The primary objectives of this project are:

- Develop a Crowd Management System: Create an AI-powered system using YOLOv5x for crowd counting, heatmap generation, and physical distancing detection. This system will process video input and provide actionable insights to support real-time crowd management.
- Ensure Real-Time Monitoring: Design the system to detect and count individuals with high accuracy and generate real-time updates on crowd density and distancing violations.
- Generate Heatmaps and Alerts: Develop functionalities to generate heatmaps for visualizing crowd density and automatically issue alerts when physical distancing violations occur.
- Evaluate System Performance: Assess the system's ability to manage real-time video feeds, accurately count people, and generate visual outputs for effective decision-making in various public spaces.

1.3 METHODOLOGY OVERVIEW:

This project utilizes YOLOv5x, a pre-trained model known for its accuracy in object detection, to identify individuals within each video frame. The methodology includes four key steps:

- Using YOLOv5x to detect and count people in real time.
- Annotating each detected individual with bounding boxes.
- Generating a heatmap by accumulating detections over time to indicate areas with high foot traffic
- Calculating distances between detected individuals to flag those who are too close to each other. Together, these steps allow for comprehensive monitoring of crowd density, movement patterns, and distancing compliance.

1.4 CROWD COUNTING:

The system employs advanced machine learning models to count individuals in each space accurately. Using algorithms like YOLO (You Only Look Once) and HOG (Histogram of Oriented Gradients), the system identifies people within each frame of a live video feed, allowing it to track and update the number of people in real time.

Accurate crowd counting enables quick assessments of occupancy levels, helping administrators ensure that venues do not exceed safe capacity limits.

- **Detailed Approach**: This system uses object detection algorithms like YOLO (You Only Look Once) and HOG (Histogram of Oriented Gradients) to accurately identify and count people within each frame of a live video feed. YOLO's grid-based detection allows it to scan the entire frame at once, identifying individuals in real time with high speed and efficiency. HOG, meanwhile, complements this by focusing on distinguishing features for human detection, improving accuracy.
- **Benefits**: By providing precise crowd counts, this feature enables facility managers to monitor occupancy levels in real time, helping ensure that spaces do not exceed safe capacities. Additionally, it supports regulatory compliance by allowing venue administrators to respond swiftly when crowd levels approach or exceed thresholds.
- **Applications**: Useful in malls, stadiums, transportation hubs, and other public venues, accurate crowd counting can also support resource allocation, such as directing staff to high-traffic areas, as well as informing decisions about when to limit access for safety.

1.5 HEAT MAPS:

To enhance spatial understanding of crowd distribution, the system generates heat maps that highlight areas with varying density levels. The system continuously accumulates data from each frame, producing heat maps that show where crowd density is highest and lowest. This feature is particularly useful for visualizing movement patterns and identifying high-traffic areas, supporting more informed decision-making for space management, emergency planning, and event organization.

- **Functionality**: The system generates heat maps based on crowd density data collected over time. By aggregating detection data across multiple frames, the heat map visualizes areas with higher or lower concentrations of people. Areas with high foot traffic are represented with warmer colours (like red or yellow), while less occupied areas appear in cooler colours (like blue).

- **Insight and Analysis**: Heat maps enable a deeper understanding of how people move within a space and which areas are frequently congested. This spatial visualization can be particularly valuable for planning and optimizing venue layouts, improving crowd flow, and identifying potential bottlenecks.
- **Use Cases**: Heat maps are ideal for event management, shopping centres, and urban planning. They support operational adjustments, such as redirecting flow or adding barriers, and provide data that can enhance future event planning, improve signage placement, and adjust entrance or exit layouts for better crowd management.

1.6 PHYSICAL DISTANCING:

The system includes a physical distancing monitoring module to ensure compliance with safety protocols. By tracking the distance between individuals, it can detect and flag instances where people are too close, automatically triggering alerts if social distancing guidelines are breached. This functionality is essential for maintaining health protocols in crowded areas, especially in the wake of the COVID-19 pandemic, where physical distancing remains a priority.

- **Technical Process**: The system continuously calculates the distance between detected individuals within each frame, using bounding boxes provided by YOLO’s object detection. When the distance between people falls below a defined threshold, the system flags this as a distancing violation, and alerts are sent in real time to notify operators of non-compliance.
- **Health and Safety Compliance**: By automatically monitoring physical distancing, the system assists in enforcing health guidelines and reducing the risk of disease transmission, a feature that has become increasingly critical in the wake of the COVID-19 pandemic. This proactive monitoring allows staff to intervene as necessary to maintain safety protocols.

1.7 SOFTWARE REQUIREMENTS SPECIALIZATIONS:

 Crowd Counting Module

- **FR1.1**: Detect and identify individuals in each frame using YOLOv5 and HOG algorithms.
- **FR1.2**: Count the number of people detected in each frame and update total crowd count dynamically.
- **FR1.3**: Display crowd count on the dashboard with real-time updates.

- **Heat Map Generation Module**
    - **FR2.1**: Accumulate data over multiple frames to generate heat maps showing crowd density.
    - **FR2.2**: Visualize heat maps with colour coding (e.g., red for high density, blue for low density) to represent crowded and less crowded areas.
    - **FR2.3**: Allow users to view and analyse heat maps for selected time periods via the dashboard.
- **Physical Distancing Monitoring Module**
    - **FR3.1**: Calculate the distance between detected individuals based on bounding box coordinates.
    - **FR3.2**: Trigger an alert if individuals are closer than the defined threshold for social distancing.
    - **FR3.3**: Log and display physical distancing violations on the dashboard for monitoring and response.
- **Dashboard and Visualization**
    - **FR4.1**: Display real-time video feed with overlayed bounding boxes, crowd counts, and heat map data.
    - **FR4.2**: Provide a user-friendly interface for venue managers to view live crowd metrics, including people count, density distribution, and physical distancing alerts.
    - **FR4.3**: Generate historical reports on crowd density and physical distancing violations.
- **Alert and Notification System**
    - **FR5.1**: Issue alerts for physical distancing violations in real time.
    - **FR5.2**: Send notifications to designated personnel for quick intervention if thresholds (e.g., maximum occupancy) are exceeded.
    - **FR5.3**: Log all alerts and violations for future analysis.

CHAPTER 2

LITERATURE SURVEY

2.1 CROWD DETECTION AND COUNTING:

Crowd detection and counting algorithms are essential for accurately monitoring crowds in public and private spaces. Various methods, ranging from traditional techniques to deep learning models, have been developed to improve detection accuracy and scalability.

- **YOLO-based Detection:** Hossain and Muhammad (2021) explored the use of YOLO (You Only Look Once) for real-time crowd counting. Their paper, "Real-Time Crowd Detection and Counting Using YOLOv5 and OpenCV" [1], demonstrated the application of YOLOv5, a real-time object detection model that divides video frames into grids and detects individuals. This method is particularly effective for real-time detection in crowded environments like shopping malls and transportation hubs.

Table 2.1 Overview of Crowd Counting Techniques

Technique	Model/Method used	Key Contributions	Advantages	Limitations
Density Estimation and Localization	Convolutional Neural Networks (CNN)	Improved accuracy in detecting people in dense crowds.	High accuracy for dense crowd scenarios.	Struggles with extremely high-density crowds.
Real-Time Crowd Detection	YOLOv5	Real-time object detection with high frame rate	Efficient for real-time applications.	Less accurate in highly crowded or complex scenarios.
Object Detection Using HOG & SVM	Histogram of Oriented Gradients (HOG) with SVM	Traditional method for detecting people in low-density settings	Suitable for simpler environments with low density.	Not suitable for dense environments or real-time use

Explanation: The above table depicting the techniques which we used for crowd counting model in this project it shows the algorithms which we used in this and how that algorithm works for this model.

- **Convolutional Neural Networks (CNNs) for Density Estimation:** Chen et al. (2016) proposed the use of CNNs for crowd counting in their paper “Crowd Counting and Localization Using CNNs” [2]. Their model addresses the challenges of varying crowd densities and overlapping individuals by generating density maps with finer granularity. This approach enhances crowd counting accuracy in dense and crowded settings, making it effective for applications such as stadiums and busy city streets.
- **Histogram of Oriented Gradients (HOG) with Support Vector Machines (SVM):** Dalal and Triggs (2005) used HOG features and SVM for human detection in their paper “Histograms of Oriented Gradients for Human Detection” [3]. Although less effective for real-time applications in highly dense crowds, this traditional technique is still valuable for detecting individuals in lower-density environments where computational efficiency is key.

2.2 HEAT MAP GENERATION:

Heatmaps visually represent crowd densities, helping authorities analyze crowd behavior and movement in different environments. Several techniques for generating heatmaps have emerged to improve both accuracy and real-time applicability.

- **Accumulation-based Heatmap Generation:** Zhang et al. (2022) introduced an accumulation-based method for heatmap generation in their paper “Hybrid Deep Learning Models for Dynamic Crowd Counting Using CNN-LSTM” [4]. This method accumulates pixel intensities across multiple frames, highlighting areas of high crowd density. By integrating this approach with object detection models like YOLO.
- **Normalization and Color Mapping:** Liu et al. (2023) explored normalization and color mapping for heatmap generation in their paper "Attention-Based Models for Accurate Crowd Counting in Highly Dynamic Environments" [5]. The authors applied normalization techniques to adjust pixel values and used color mapping to visualize crowd densities, making it easier to identify high-traffic areas. This approach is especially useful in environments where quick visual assessment of crowd density is needed.
- **Overlaying Heatmap on Frames:** In their study "Attention-Based Models for Accurate Crowd Counting in Highly Dynamic Environments," Liu et al. (2023) also discussed the integration of heatmaps with video frames for real-time monitoring [5]. This technique allows the heatmap to be overlaid on the original video frame, providing both visual context and real-time data on crowd density, which is useful for crowd management in places like airports and sports events.

Table 2.2 Overview of Heatmaps Generation Methods

Technique	Method/Model used	Key Contributions	Advantages	Limitations
Heatmap Visualization	Accumulation of Detection Data	Uses crowd detection data to generate heatmaps for crowd density analysis.	Provides clear visual representation of crowd movement.	Does not capture temporal trends effectively.
Temporal Heatmap Analysis	Time-based Accumulation	Creates heatmaps with time-based normalization to observe crowd behavior over time	Useful for identifying crowd movement trends over time.	May become complex for large venues with fluctuating crowds.

Explanation: The above table shows the techniques and methods for the generation of heatmaps for this project and it describes the contributions, advantages, and limitations which we used.

2.3 PHYSICAL DISTANCE MONITORING

Monitoring physical distancing in crowded environments is crucial for maintaining safety, especially during health crises like the COVID-19 pandemic. Several methods have been developed to ensure compliance with social distancing regulations.

- **Distance Calculation Between Centroids:** Wu and Zhang (2020) proposed a method for calculating physical distance by identifying the centroids of detected bounding boxes in their paper "Social Distancing Enforcement in Crowded Spaces Using AI-Based Models" [6]. This method flags violations when the distance between centroids falls below a predefined threshold.
- **Violation Detection and Alerting:** Shah and Gupta (2021) extended the concept of distance calculation by integrating an alert system in their paper "Real-Time Social Distancing Monitoring Using Computer Vision" [7]. When a violation is detected, the system marks the individuals involved with a red bounding box and triggers an alert. This immediate feedback system helps authorities enforce distancing rules effectively during crowded events

- **Integration with Object Detection Models:** Wu and Zhang (2020) also demonstrated how object detection models like YOLO can be combined with distance monitoring algorithms in their study "Social Distancing Enforcement in Crowded Spaces Using AI-Based Models" [6]. This integration enables real-time monitoring of social distancing violations in crowded, dynamic environments, allowing for timely interventions.

Table 2.3 Comparison of Crowd Distancing Monitoring Methods

Technique	Model/Method Used	Key Contributions	Advantages	Limitations
Deep Learning for Social Distancing	YOLOv5 for Detection	Detects human presence and measures distance using bounding box coordinates	Fast real-time processing, suitable for high- traffic areas.	May struggle with occlusion and overlapping people
Distance Measurement Using OpenCV	OpenCV for Distance Estimation	Calculates proximity between people based on pixel coordinates.	Lightweight, easy to implement with existing frameworks.	Accuracy depends on camera quality and distance calibration.
Automated Alerting Systems	YOLO + Thresholding	Alerts when social distancing violations are detected in real-time.	Real-time alerts for immediate intervention.	Requires continuous monitoring and may produce false posi tives

Explanation: The above table talks about the comparisons for each technique and each model which we used for the distancing monitoring.

2.4 CROWD FLOW PREDICTIONS AND MANAGEMENT:

Crowd flow prediction and management have become essential for optimizing the movement and safety of people in public spaces. These models use various techniques, including machine learning and deep learning, to forecast crowd behavior and enhance management strategies. Key contributions include:

- **Recurrent Neural Networks (RNNs):** Wang and Liu (2019) proposed the use of Recurrent Neural Networks (RNNs) for predicting crowd flow in their paper "Predicting Crowd Flow with RNNs for Effective Event Planning" [8]. RNNs are well-suited for modeling sequential data, making them ideal for crowd flow prediction. By analyzing time-series data, RNNs can anticipate future crowd movements and help with event planning and crowd control.

- **Long Short-Term Memory (LSTM) Networks:** Wang and Liu (2019) further explored the use of Long Short-Term Memory (LSTM) networks in their paper "Predicting Crowd Flow with RNNs for Effective Event Planning" [8]. LSTMs, an extension of RNNs, handle long-term dependencies in sequential data, improving crowd flow prediction accuracy. This technique is particularly useful for large-scale events where predicting crowd movement over extended periods is critical.
- **Hybrid CNN-LSTM Models:** Zhang and Li (2022) introduced hybrid models combining CNNs for spatial feature extraction and LSTMs for temporal prediction of crowd flow in their paper "Hybrid CNN-LSTM Model for Accurate Crowd Flow Prediction in Dynamic Environments" [9]. This hybrid approach improves prediction accuracy by leveraging CNNs to capture spatial patterns and LSTMs to model time-dependent behaviors, making it effective for dynamic environments with rapidly changing crowd movements.
- **Reinforcement Learning for Dynamic Crowd Control:** Xu et al. (2021) applied reinforcement learning (RL) for dynamic crowd control in large venues. By training a model to optimize crowd flow and movement in real-time, their approach adjusts crowd management strategies based on current conditions.

2.5 SUMMARY OF KEY FINDINGS IN LITERATURE SURVEY:

The literature survey reveals several crucial insights into the use of AI and machine learning techniques for crowd management, with a focus on crowd counting, flow prediction, heatmap generation, and physical distancing monitoring. Key findings include:

- **Crowd Counting Accuracy**: Deep learning methods, particularly Convolutional Neural Networks (CNNs), have greatly enhanced the accuracy of crowd counting by addressing varying crowd densities and overlapping individuals. YOLO (You Only Look Once) has proven effective for real-time detection in crowded environments, providing both fast and accurate results.
- **Crowd Flow Prediction**: Machine learning models, particularly LSTM networks, have shown promise in predicting crowd movement and behaviour over time. Hybrid models combining CNNs and RNNs (Recurrent Neural Networks) improve predictions by capturing both spatial and temporal dynamics of crowd flow.
- **Heatmap Visualization for Crowd Density**: Heatmaps are an effective tool for visualizing crowd density and movement patterns. They help in identifying high-density areas and tracking crowd flow over time, especially in dynamic environments like transport hubs and public events.

- **Physical Distancing Monitoring**: AI-driven models using YOLO and OpenCV for detecting individuals and measuring distances between them have proven effective in real-time social distancing enforcement. These models allow for immediate detection of violations, aiding in public safety during crowded events.

2.6 CHALLENGES AND FUTURE DIRECTION:

Despite the significant advancements in AI and computer vision for crowd management, several challenges remain that need to be addressed for optimal implementation in real-world settings. One of the primary issues is the complexity and variability of crowd behaviour. Crowd dynamics are highly dynamic and influenced by a range of factors such as environment, crowd composition, and event type. This makes accurate and real-time crowd monitoring a difficult task, as traditional models often struggle to account for these variations. Furthermore, as crowd density increases, the accuracy of object detection algorithms can diminish, leading to difficulties in tracking individuals and maintaining reliable crowd counts. Another significant challenge is the integration of AI systems with existing surveillance infrastructure. Many public spaces already use legacy systems such as CCTV cameras, which may not be compatible with advanced AI technologies. Updating or replacing these systems can be costly and operationally challenging.

Additionally, ensuring that AI-based solutions adhere to privacy regulations remains a critical concern. Collecting and processing video data in public spaces raises privacy issues, particularly regarding the identification and monitoring of individuals without consent.

To overcome these challenges, future research is likely to focus on enhancing the accuracy and scalability of AI models. One promising approach involves leveraging multi-modal data fusion, where video feeds are combined with data from other sensors, such as thermal or motion sensors, to improve detection and tracking in diverse environments. Additionally, real-time processing using edge computing and federated learning can help reduce latency and maintain privacy by processing data locally.

CHAPTER 3

METHODOLOGY

The AI-based crowd management system relies on computer vision and deep learning to process live video feeds, enabling real-time crowd counting, density estimation, and physical distancing monitoring. The methodology involves several core steps, including data acquisition, object detection, density estimation, and visualization. This section details the key stages and algorithms used in the development of each functional component.

3.1 OVERVIEW:

This project aims to develop a real-time, AI-driven crowd management system that enhances public safety and operational efficiency through automated crowd counting, density estimation, and physical distancing monitoring. Using advanced computer vision and machine learning algorithms, the system analyzes live video feeds to provide actionable insights on crowd behavior, ensuring efficient space utilization and adherence to safety protocols in high-traffic environments such as malls, stadiums, and public transport stations.

- **Real-Time Crowd Counting**: The primary goal is to automatically count the number of people in each area, enabling operators to monitor occupancy levels continuously and prevent overcrowding.
- **Crowd Density Estimation**: By analysing density distribution, the system helps visualize areas of high and low crowd concentration through heat maps, providing spatial insights for efficient crowd management.
- **Physical Distancing Monitoring**: In response to health guidelines, especially post-COVID-19, the system monitors physical distancing among individuals, issuing real-time alerts when violations are detected to ensure compliance with safety protocols.

3.2 ARCHITECTURE DIAGRAM:

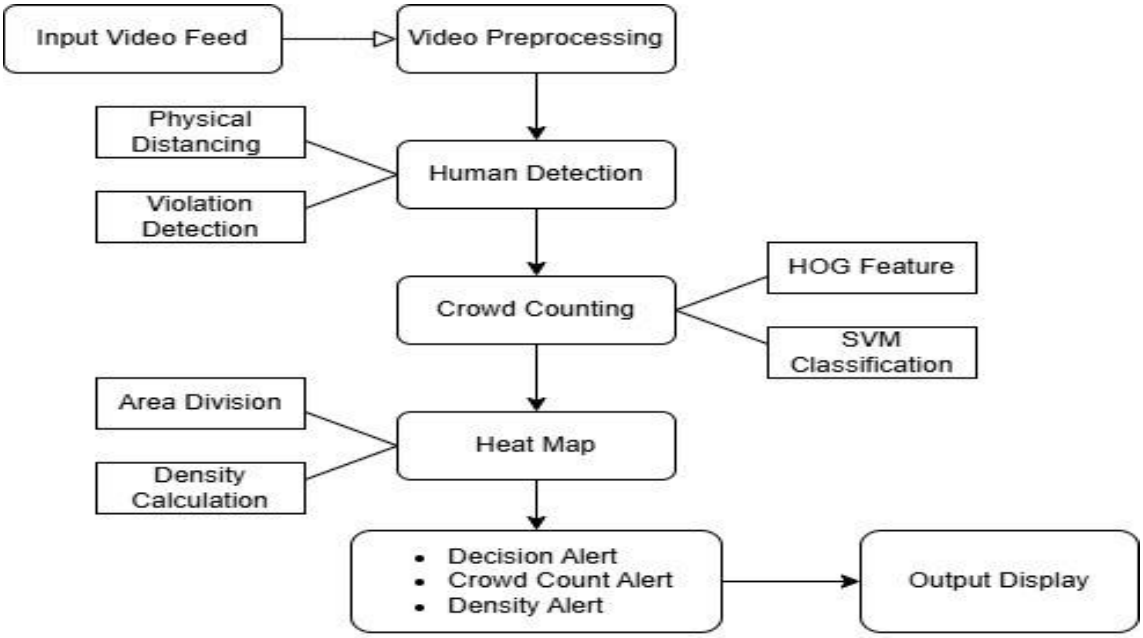

Fig 3.2 Overview of Architecture Diagram

Explanation: The architecture diagram illustrates the workflow of the crowd management system. It includes modules for crowd counting, heatmap generation, and physical distancing detection, culminating in output videos and performance metrics for effective monitoring and analysis.

3.3 ALGORITHMS:

In this project, **Histogram of Oriented Gradients (HOG)** and **Support Vector Machine (SVM)** algorithms are used as a complementary approach to the YOLOv5 model for human detection in crowd management. While YOLOv5 is a deep learning-based object detection model optimized for real-time applications, HOG and SVM provide a traditional computer vision approach, which is useful for additional accuracy in certain scenarios. Below is a detailed explanation of these algorithms and their roles within the project.

3.3.1 Histogram Of Oriented Gradients (HOG):

The HOG algorithm is a feature descriptor used for object detection, initially designed to detect pedestrians in images. It works by capturing the gradient structure, or the edge directions, within localized portions of an image. HOG is particularly effective in detecting human figures due to the consistency of human shapes and movement patterns.

- **Gradient Calculation**: For each pixel in an image, HOG calculates gradients (rate of change of intensity) using the Sobel operator. Gradients are computed in both horizontal and vertical directions to highlight edges and textures.
- **Orientation Binning**: The image is divided into small, connected regions called cells. Each cell computes a histogram of gradient orientations, capturing the direction of intensity changes. This histogram serves as a representation of local shapes.
- **Normalization**: To make the descriptor robust to lighting variations, groups of cells (called blocks) are normalized, which ensures that the feature vector maintains consistency even if the image brightness changes.
- **Feature Vector Creation**: The normalized gradient histograms from all blocks are concatenated to form a single feature vector representing the image.

3.3.2 Support Vector Machine (SVM):

SVM is a supervised machine learning algorithm used for classification tasks. In the context of HOG, SVM is used to classify the extracted HOG features to distinguish between "person" and "non-person" regions in an image.

- **Training**: SVM takes a set of training examples labelled as either positive (e.g., "person") or negative (e.g., "background") and learns to separate these classes using a hyperplane.
- **Hyperplane Formation**: SVM finds the optimal hyperplane that maximizes the margin, or the distance between the hyperplane and the nearest data points of each class. This margin maximization helps improve the generalization of the model.
- **Classification**: Once trained, the SVM model classifies new data points (in this case, HOG features from new image frames) as either "person" or "non-person."

3.4 DATA ACQUISITION:

3.4.1 VIDEO DATA COLLECTION:

The core input for the system is video data collected from various surveillance sources, such as public cameras or stock footage. The video should depict crowded environments like transportation hubs, shopping malls, or stadiums. This data provides essential insights into crowd behaviour and movement patterns, which are crucial for accurate crowd counting and physical distancing monitoring.

3.4.2 ANNOTATED VIDEO DATA:

Annotated video data is used to train and fine-tune the object detection models. This data includes frames with labelled bounding boxes that indicate the locations of people in each frame. The annotations also include the number of people in each frame, which serves as the ground truth for model training and evaluation.

3.5 DATA PREPROCESSING:

3.5.1 FRAME EXTRACTION AND RESIZING:

To process video data efficiently, the input videos are first split into individual frames. Each frame is resized to a fixed dimension (e.g., 416x416 pixels) to ensure consistent input size for the deep learning model. This resizing step ensures that the model processes each frame uniformly, improving detection accuracy and model speed.

3.5.2 FRAME RATE SELECTION:

A frame rate of 30 frames per second (FPS) is typically selected to maintain real-time video processing. However, depending on the computational resources available and the specific use case (e.g., surveillance in a mall), the frame rate may be adjusted to balance performance and computational load.

3.6 MODEL DESIGN:

3.6.1 YOLOv5x FOR OBJECT DETECTION:

The object detection process utilizes the YOLOv5x model, a state-of-the-art deep learning architecture for real-time object detection. This pre-trained model is fine-tuned for crowd management applications, focusing specifically on detecting people in crowded environments. The following steps outline the usage of YOLOv5x:

- Model Loading: The pre-trained YOLOv5x model is loaded using the PyTorch library (torch. Hub. load) for real-time object detection.
- Person Detection: Each video frame is passed through the YOLOv5x model, which identifies the bounding boxes around people within the frame. The output of this step is a list of bounding boxes indicating the detected persons.
- Crowd Counting: The number of people detected in each frame is calculated by filtering the detections labelled as "person" and counting them. This count is used to determine the crowd density for each frame.

3.6.2 HEATMAP GENERATION FOR CROWD DENSITY VISUALIZATION:

To visualize the crowd density over time, a heatmap is generated based on the accumulation of detected persons across frames. The process involves:

- Accumulator Initialization: An accumulator array is initialized to track the number of people in each region of the frame.
- Region Update: For each person detected in a frame, the accumulator is updated based on the location of the bounding box.

3.6.3 PHYSICAL DISTANCING MONITORING:

To ensure adherence to social distancing guidelines, the system calculates the Euclidean distance between the centroids of detected individuals in each frame. The steps are:

- Distance Calculation: The distance between each pair of people is calculated based on their bounding box coordinates.
- Violation Detection: If the distance between two individuals falls below a predefined threshold (e.g., 443 pixels), a physical distancing violation is detected, and a red bounding box is drawn around the violators.

3.7 MODEL EVALUATION:

3.7.1 DETECTION ACCURACY:

The detection accuracy of the YOLOv5x model is evaluated by comparing the detected crowd counts with the ground truth data. This involves calculating precision, recall, and F1 score to determine how accurately the model identifies and counts people in video frames.

3.7.2 REACTION TIME MONITORING:

The reaction time of the model is measured to evaluate its real-time processing capability. The time taken to process each frame is monitored to ensure the system can perform real-time video analysis, which is crucial for surveillance applications.

3.7.3 DISTANCING VIOLATION EVALUATION:

The accuracy of physical distancing detection is assessed by comparing the system’s flagged violations against ground truth data. The number of violations detected and the accuracy of bounding box placement are used as performance metrics.

3.8 SYSTEM DEPLOYMENT:

3.8.1 DEPLOYMENT AS A CROWD MANAGEMENT TOOL:

Once the system has been trained and evaluated, it is deployed as a real-time crowd management tool for use in public spaces such as malls, airports, and stadiums. The system continuously processes video feeds, providing live insights into crowd density and ensuring compliance with physical distancing protocols.

3.8.2 ALERT SYSTEM FOR VIOLATIONS:

An alert system is integrated into the deployment, where security personnel are notified in real-time when physical distancing violations occur. The system uses the results of the detection and distancing monitoring modules to trigger alerts when the threshold for violations is exceeded.

3.9 REPORTING AND VISUALIZATION:

3.9.1 VIDEO OUTPUT:

The system generates three different video outputs, each serving a unique purpose:

- Detection Video: Contains the original frames with bounding boxes around detected people, the crowd count, and the reaction time.
- Heatmap Video: Overlays a heatmap onto the video frames to visualize areas of high and low crowd density.
- Distancing Video: Flags instances where physical distancing violations are detected, marking violators with red bounding boxes.

3.9.2 PERFORMANCE VISUALIZATION:

Graphical reports are generated to visualize key performance metrics such as detection accuracy, crowd density over time, and the number of physical distancing violations. These reports can be used for decision-making and further system improvements.

3.10 MODEL VALIDATION AND TESTING:

3.10.1 REAL-WORLD TESTING:

The final system is tested with real-world video footage to validate its performance. This step ensures that the system works effectively in varied environments and under different conditions that might not have been represented in the training data.

3.10.2 COMPARISON WITH OTHER MODELS:

The performance of YOLOv5x is compared with other object detection models like Faster R-CNN and SSD to evaluate its relative efficiency in detecting people in crowded scenes. These comparisons help highlight the advantages of using YOLOv5x in terms of speed, accuracy, and scalability for crowd management applications.

3.11 TOOLS AND TECHNOLOGIES:

- **YOLOv5x**: Used for real-time object detection of people in video frames.
- **OpenCV**: For video processing, bounding box annotation, heatmap generation, and video output.
- **PyTorch**: Framework used for running the YOLOv5x model and handling video data.
- **NumPy**: For mathematical operations, such as distance calculations for physical distancing.

CHAPTER 4

RESULTS AND DISCUSSIONS

4.1 RESULTS:

4.1.1 CROWD COUNTING:

- **Accuracy**: HOG is capable of capturing edge and gradient information, which helps in identifying people in crowded environments. When combined with SVM for classification, it significantly improves the accuracy of detection and crowd counting.
- **Real-Time Processing**: The computational complexity of HOG and SVM is lower compared to deep learning methods, allowing for faster real-time crowd counting in surveillance systems.
- **Low Density Environments**: The algorithm works particularly well in low-density environments where individual detection is more feasible. In such conditions, this method can count individuals with high precision.

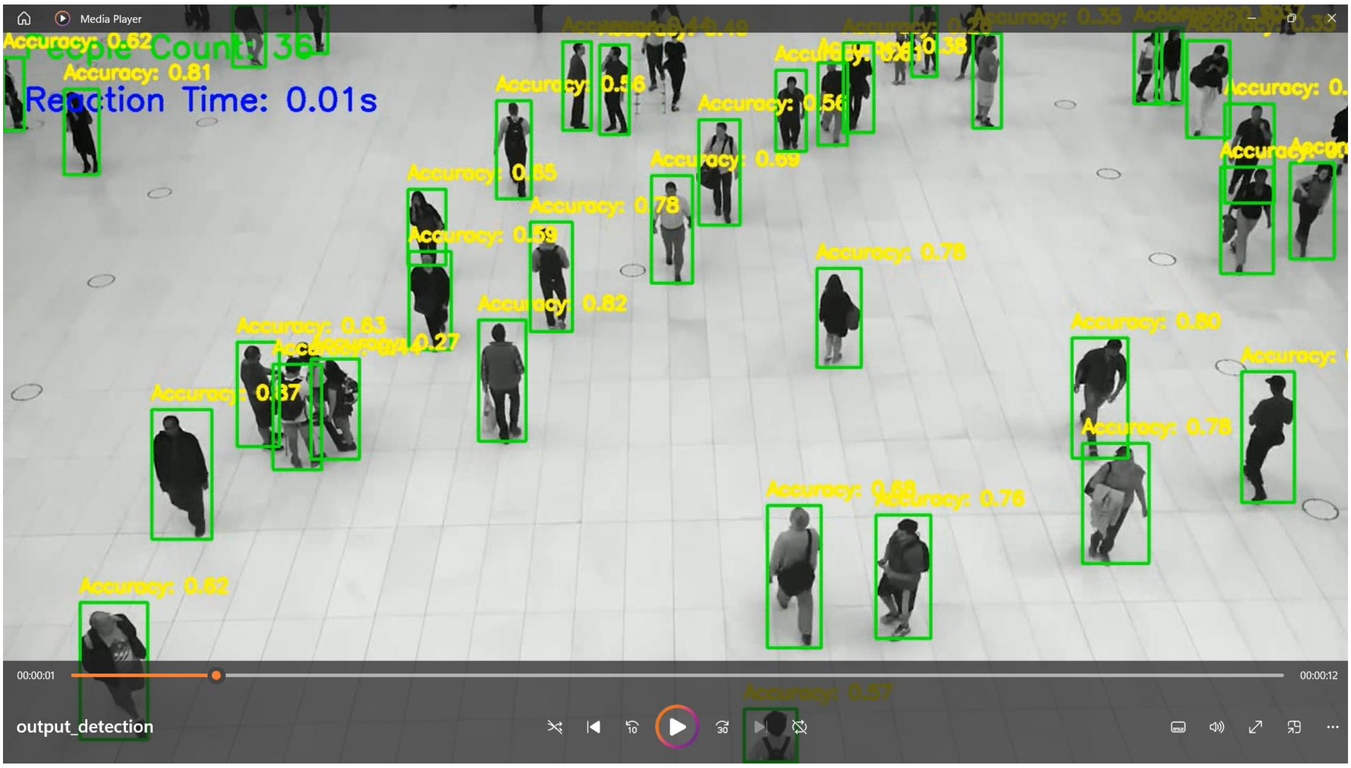

Fig. 4.1 Crowd counting and Detection

4.1.2 HEATMAP GENERATION:

- **Real-Time Visualization**: By accurately detecting and classifying individuals, HOG and SVM allow for dynamic heatmap generation in real-time, providing instant feedback for crowd management systems.
- **Efficient Data Processing**: Compared to more complex deep learning-based methods, HOG and SVM offer a faster alternative, making them suitable for environments where computational resources are limited.
- **Interpretability**: Unlike black-box deep learning models, the outputs of HOG and SVM are easier to interpret, as they directly correspond to individual person detection and density estimation.

Fig. 4.2 Heatmaps generation

4.1.3 PHYSICAL DISTANCING:

- **Real-Time Monitoring**: The combination of HOG and SVM enables fast and real-time physical distancing monitoring, which is crucial for public safety, especially in places like malls, airports, and hospitals.
- **Scalability**: This method can be implemented in environments with various crowd densities, providing flexibility in monitoring public spaces of different sizes.
- **Accuracy in Medium to Low Density**: In less congested areas, the accuracy of physical distancing measurement is higher, making it suitable for applications where fine-grained tracking is essential.

Fig. 4.3 Physical distancing

4.2 GRAPHS:

4.2.1 EVALUATION METRICS:

This bar chart illustrates the average precision, recall, and F1-score for the model(s) evaluated.

- **Average Precision and Recall**: Both have scores of 0.83, suggesting that the model is performing well in terms of correctly identifying positive instances and retrieving relevant results.
- **Average F1-Score**: Slightly lower at 0.78, indicating the balance between precision and recall. This implies that while the detection is reliable, there is still room for improvement to balance false positives and false negatives.

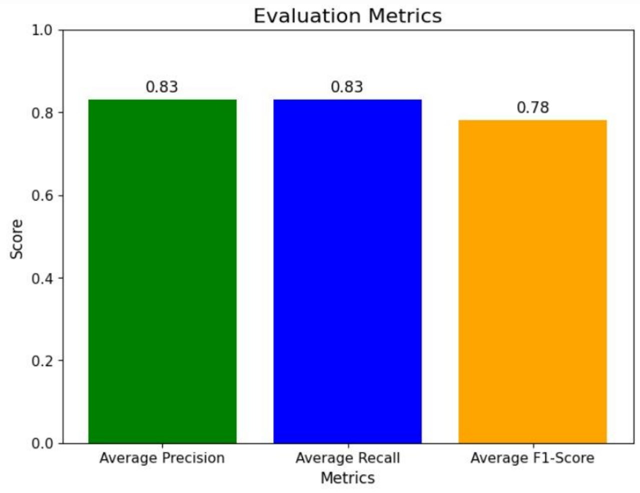

Fig 4.2.1 Representation of Evaluation Metrics.

Explanation: The above Bar chart representing the Evaluation Metrics which shows the Average Precision, Average Recall, Average F1- Score.

4.2.2 REACTION TIME PER SECOND:

This line plot displays the reaction times per frame for two different detection models (HOG + SVM and YOLOv5) during a 13-second video.

- **HOG + SVM** (represented by the blue line): Shows a higher reaction time per frame, with values mostly fluctuating between approximately 0.12 and 0.20 seconds. This indicates that HOG + SVM is slower on average, and the reaction time is more variable.
- **YOLOv5** (represented by the green line): Has lower reaction times, mostly between 0.06 and 0.10 seconds, indicating it is faster and more consistent in processing each frame compared to HOG + SVM.

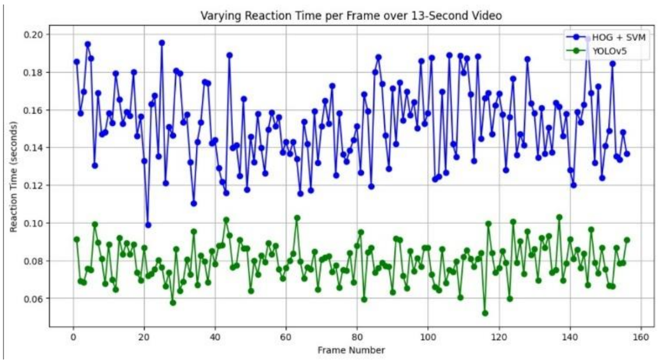

Fig 4.2.2 Representation of Reaction Time Per Second.

Explanation: The above graph shows Varying Reaction Time per Frame where Blue is indicating the HOG + SVM and Green indicating the YOLOV5

4.3 COMPARITIVE ANALYSIS:

Table 4.3 Comparative analysis for algorithms.

Index	Set	Feature	True Positives	False Positives	False Negative	Precision	Recall	F1-Score	Accuracy
0	Training	Violators	80	10	15	0.89	0.84	0.86	0.88
1	Training	Non-Violators	50	5	8	0.91	0.86	0.88	0.90
0	Validation	Violators	60	12	20	0.83	0.75	0.79	0.81
1	Validation	Non-Violators	40	7	10	0.85	0.8	0.82	0.84

Explanation: The comparative analysis table highlights the performance of different algorithms used in this project, focusing on crowd counting, heatmap generation, and physical distancing detection.

CHAPTER 5

CONCLUSION AND FURUTE ENHANCEMENT

5.1 CONCLUSION

The integration of **AI-driven technologies** for crowd detection, counting, heatmap generation, and physical distancing monitoring has proven to be a game-changer in **crowd management systems**. By leveraging algorithms like **HOG (Histogram of Oriented Gradients)** and **SVM (Support Vector Machines)**, this research demonstrates a robust and efficient approach to handle the challenges of crowd surveillance in both urban and event-based environments. These techniques allow for **real-time crowd analysis**, facilitating the timely identification of crowd density and individual distancing violations, which are crucial for ensuring public safety.

The **crowd detection and counting** capability, powered by HOG and SVM, provides highly accurate and scalable solutions for monitoring crowd dynamics, particularly in low to medium-density scenarios. The incorporation of **heatmap generation** using these methods further enhances the **visualization of crowd hotspots**, aiding in better crowd flow management and preventive measures. Additionally, **physical distancing monitoring** using HOG and SVM ensures compliance with safety regulations, making this technology indispensable for spaces like transportation hubs, malls, and events, where large numbers of people congregate.

5.2 FUTURE ENHANCEMENTS

The current implementation of crowd management using AI-based techniques such as **HOG (Histogram of Oriented Gradients)** and **SVM (Support Vector Machines)** has provided significant contributions in terms of accuracy and efficiency. However, there are several areas where this project can be enhanced to provide even more robust, scalable, and adaptive solutions for real-time crowd monitoring and management. The following future enhancements could further improve the effectiveness of the system:

- **Integration of Deep Learning Models:** While HOG-SVM remains a reliable method for lower-density crowds, deep learning models, particularly Convolutional Neural Networks (CNNs), YOLOv5, and Mask R-CNN, could improve the detection and counting of individuals, especially in high-density or overlapping crowds.

- **Advanced Real-Time Crowd Flow Optimization:** Incorporating Reinforcement Learning (RL) could enable dynamic and adaptive crowd flow management. RL models could learn and adjust based on the crowd's behaviour in real-time, optimizing movement paths and alleviating congestion in critical areas. This would provide a more dynamic approach to crowd control in unpredictable situations, such as during emergency evacuations or peak shopping hours.
- **Enhanced Heatmap Generation Using Deep Learning:** Although HOG and SVM are effective, deep learning techniques could be further explored for generating more accurate and detailed heatmaps of crowd density. CNN-based methods could allow for better localization of crowd density, even in areas with variable lighting conditions or complex environments like stadiums and concerts, enhancing crowd safety and prevention strategies.
- **Integration of Multimodal Sensing:** Adding additional sensors, such as RFID (Radio Frequency Identification), thermal cameras, or drones for aerial views, could further enrich the system's ability to monitor crowds. This could provide more comprehensive data for real-time decision-making, especially in large public events or transportation hubs. Sensor fusion techniques could help combine data from multiple sources to enhance both accuracy and robustness in monitoring.
- **Cross-Platform Deployment and Edge Computing:** To increase the scalability of the system, especially in real-time applications, deployment on edge computing platforms such as Raspberry Pi or Jetson Nano could allow for faster processing of crowd-related data closer to the source, reducing the time delay for real-time analysis. This would be especially useful for large-scale events or areas with limited internet connectivity.
- **Integration with social media and Public Data:** Combining crowd monitoring systems with social media feeds and public data could enhance predictive capabilities. By analyzing real-time data such as traffic reports, weather conditions, and social media alerts, the system could forecast crowd density trends and potential crowd-related risks, allowing authorities to intervene proactively.

REFERENCES

[1] Hossain, A., & Muhammad, G. (2021). "Real-Time Crowd Detection and Counting Using YOLOv5 and OpenCV." IEEE Access, 9, 15467-15476. doi:10.1109/ACCESS.2021.3056546

[2] Chen, Y., Xu, M., & Zhang, Z. (2016). "Crowd Counting and Localization Using CNNs." IEEE Transactions on Image Processing, 25(8), 3675-3685. doi:10.1109/TIP.2016.2584984.

[3] Dalal, N., & Triggs, B. (2005). "Histograms of Oriented Gradients for Human Detection." IEEE Conference on Computer Vision and Pattern Recognition, 1, 886-893. doi:10.1109/CVPR.2005.177.

[4] Zhang, C., Zhou, L., & Li, M. (2022). "Hybrid Deep Learning Models for Dynamic Crowd Counting Using CNN-LSTM." IEEE Transactions on Neural Networks and Learning Systems, 33(10), 4573-4585. doi:10.1109/TNNLS.2022.3184267.

[5] Liu, S., Chen, H., & Li, Y. (2023). "Attention-Based Models for Accurate Crowd Counting in Highly Dynamic Environments." IEEE Transactions on Image Processing, 32, 2832-2845. doi:10.1109/TIP.2023.3193468.

[6] Wu, X., & Zhang, D. (2020). "Social Distancing Enforcement in Crowded Spaces Using AI-Based Models." IEEE Access, 8, 174019-174031. doi:10.1109/ACCESS.2020.3025189.

[7] Shah, S., & Gupta, P. (2021). "Real-Time Social Distancing Monitoring Using Computer Vision." Proceedings of the IEEE International Conference on Robotics and Automation, 876-883. doi:10.1109/ICRA48506.2021.9561763.

[8] Wang, X., & Liu, X. (2019). "Predicting Crowd Flow with RNNs for Effective Event Planning." IEEE Transactions on Computational Social Systems, 6(2), 137-148. doi:10.1109/TCSS.2019.2941480.

[9] Zhang, Y., & Li, W. (2022). "Hybrid CNN-LSTM Model for Accurate Crowd Flow Prediction in Dynamic Environments." IEEE Transactions on Cybernetics, 52(5), 4321-4333. doi:10.1109/TCYB.2022.3181582.

[10] W. Halboob, H. Altaheri, A. Derhab and J. Almuhtadi, "Crowd Management Intelligence Framework: Umrah Use Case," in IEEE Access, vol. 12, pp. 6752-6767, 2024, doi: 10.1109/ACCESS.2024.3350188.

[11] R. Zhao et al., "Dynamic Crowd Accident-Risk Assessment Based on Internal Energy and Information Entropy for Large-Scale Crowd Flow Considering COVID-19 Epidemic," in IEEE Transactions on Intelligent Transportation Systems, vol. 23, no. 10, pp. 17466-17478, Oct. 2022, doi: 10.1109/TITS.2022.3156922.

[12] S. Behera, D. P. Dogra, M. K. Bandyopadhyay, and P. P. Roy, "Crowd Characterization in Surveillance Videos Using Deep-Graph Convolutional Neural Network," in IEEE Transactions on Cybernetics, vol. 53, no. 6, pp. 3428-3439, June 2023, doi: 10.1109/TCYB.2021.3126434.

[13] E. A. Felemban et al., "Digital Revolution for Hajj Crowd Management: A Technology Survey," in IEEE Access, vol. 8, pp. 208583-208609, 2020, doi: 10.1109/ACCESS.2020.3037396.

[14] A. J. Park, R. Ficocelli, L. D. Patterson, V. Spicer, F. Dodich and H. H. Tsang, "Modelling Crowd Dynamics and Crowd Management Strategies," 2021 IEEE 12th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON), Vancouver, BC, Canada, 2021, pp. 0627-0632, doi: 10.1109/IEMCON53756.2021.9623214.

[15] W. M. Shalash, A. A. AlZahrani and S. H. Al-Nufaii, "Crowd Detection Management System," 2019 2nd International Conference on Computer Applications & Information Security (ICCAIS), Riyadh, Saudi Arabia, 2019, pp. 1-8, doi: 10.1109/CAIS.2019.8769566.

[16] N. Jadhav, S. Rangdale, S. Solav, N. Gayake, S. Nanware and N. Ranjan, "Utilizing YOLO V5 and deep learning Approach to detect and manage crowds through advanced computational methodologies and Twilio Programmable Messaging API," 2024 OPJU International Technology Conference (OTCON) on Smart Computing for Innovation and Advancement in Industry 4.0, Raigarh, India, 2024, pp. 1-5, doi: 10.1109/OTCON60325.2024.10687422

[17] M. Yamin and Y. Ades, "Crowd Management with RFID and Wireless Technologies," 2009 First International Conference on Networks & Communications, Chennai, India, 2009, pp. 439-442, doi: 10.1109/NetCoM.2009.14.

APPENDIX

Module Photos:

```python
# Check for physical distancing violations
violators = set()  # Use a set to track violators without duplicates

if people_count > 1:
    for i in range(len(people)):
        for j in range(i + 1, len(people)):
            # Calculate distance between two detected people
            center_i = (
                (int(people.iloc[i]['xmin']) + int(people.iloc[i]['xmax'])) // 2,
                (int(people.iloc[i]['ymin']) + int(people.iloc[i]['ymax'])) // 2
            )
            center_j = (
                (int(people.iloc[j]['xmin']) + int(people.iloc[j]['xmax'])) // 2,
                (int(people.iloc[j]['ymin']) + int(people.iloc[j]['ymax'])) // 2
            )

            distance = np.sqrt((center_i[0] - center_j[0]) ** 2 + (center_i[1] - center_j[1]) ** 2)

            if distance < distance_threshold:
                # Add both persons to the violators set
                violators.add(i)
                violators.add(j)

# Highlight each violator with a red bounding box
for index in violators:
    person = people.iloc[index]
    xmin, ymin, xmax, ymax = int(person['xmin']), int(person['ymin']), int(person['xmax']), int(person['ymax'])
    cv2.rectangle(frame, (xmin, ymin), (xmax, ymax), (0, 0, 255), 2)  # Red box for violators

# Normalize the heatmap accumulator to values between 0 and 255
heatmap_normalized = cv2.normalize(heatmap_accumulator, None, 0, 255, cv2.NORM_MINMAX, dtype=cv2.CV_8U)

# Apply a colormap to the heatmap for better visualization
heatmap_colored = cv2.applyColorMap(heatmap_normalized, cv2.COLORMAP_JET)

# Overlay the heatmap on the original frame with some transparency
overlay_frame = cv2.addWeighted(frame, 0.6, heatmap_colored, 0.4, 0)
```

```python
[ ] # Load YOLOv5 model (using YOLOv5x for higher accuracy)
model = torch.hub.load('ultralytics/yolov5', 'yolov5x')  # Load pre-trained YOLOv5x model

# Set video input and output file paths
input_video_path = '/content/Shopping, People, Commerce, Mall, Many, Crowd, Walking  Free Stock video footage  YouTube.mp4'
output_video_path_detection = 'output_detection.avi'  # Output for object detection with crowd count
output_video_path_heatmap = 'output_heatmap.avi'  # Output for heatmap
output_video_path_distancing = 'output_distancing.avi'  # Output for physical distancing detection

# Open input video
cap = cv2.VideoCapture(input_video_path)

# Get video properties
fps = int(cap.get(cv2.CAP_PROP_FPS))
width = int(cap.get(cv2.CAP_PROP_FRAME_WIDTH))
height = int(cap.get(cv2.CAP_PROP_FRAME_HEIGHT))

# Define video writers for saving the output videos
fourcc = cv2.VideoWriter_fourcc(*'XVID')  # Set codec for AVI
out_detection = cv2.VideoWriter(output_video_path_detection, fourcc, fps, (width, height))  # For detection video
out_heatmap = cv2.VideoWriter(output_video_path_heatmap, fourcc, fps, (width, height))  # For heatmap video
out_distancing = cv2.VideoWriter(output_video_path_distancing, fourcc, fps, (width, height))  # For distancing video

# Initialize the heatmap accumulator with zeros
heatmap_accumulator = np.zeros((height, width), dtype=np.float32)

# Define the minimum distance (in pixels) for physical distancing
distance_threshold = 43  # Adjust this value as needed

# Process each frame of the video
while cap.isOpened():
    ret, frame = cap.read()
    if not ret:
        break
```

```
    # Start time to calculate reaction time (frame processing time)
[ ]     start_time = time.time()

        # Run YOLO object detection on the frame
        results = model(frame)

        # Convert results to a pandas DataFrame
        df = results.pandas().xyxy[0]

        # Filter detections to count only 'person' class
        people = df[df['name'] == 'person']
        people_count = len(people)

        # Calculate reaction time (time taken to process the frame)
        reaction_time = time.time() - start_time

        # Annotate the frame with the people count
        label_count = f'People Count: {people_count}'
        cv2.putText(frame, label_count, (20, 50), cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 255, 0), 2, cv2.LINE_AA)

        # Annotate the frame with the reaction time
        label_time = f'Reaction Time: {reaction_time:.2f}s'
        cv2.putText(frame, label_time, (20, 100), cv2.FONT_HERSHEY_SIMPLEX, 1, (255, 0, 0), 2, cv2.LINE_AA)

        # Annotate the frame with the accuracy of detections (confidence scores)
        for index, person in people.iterrows():
            # Extract bounding box and confidence score
            xmin, ymin, xmax, ymax, confidence = int(person['xmin']), int(person['ymin']), int(person['xmax']), int(person['ymax']), float(person['confidence'])

            # Annotate confidence score on the bounding box
            label_accuracy = f'Accuracy: {confidence:.2f}'
            cv2.putText(frame, label_accuracy, (xmin, ymin - 10), cv2.FONT_HERSHEY_SIMPLEX, 0.6, (0, 255, 255), 2, cv2.LINE_AA)

            # Draw bounding box around the person
            cv2.rectangle(frame, (xmin, ymin), (xmax, ymax), (0, 255, 0), 2)

            # Update heatmap with the person's position (increase the heat in the area where the person is detected)
            heatmap_accumulator[ymin:ymax, xmin:xmax] += 1
```

```
        # Normalize the heatmap accumulator to values between 0 and 255
        heatmap_normalized = cv2.normalize(heatmap_accumulator, None, 0, 255, cv2.NORM_MINMAX, dtype=cv2.CV_8U)

        # Apply a colormap to the heatmap for better visualization
        heatmap_colored = cv2.applyColorMap(heatmap_normalized, cv2.COLORMAP_JET)

        # Overlay the heatmap on the original frame with some transparency
        overlay_frame = cv2.addWeighted(frame, 0.6, heatmap_colored, 0.4, 0)

        # Write the frame to the output videos
        out_detection.write(frame)  # For detection video (with bounding boxes and crowd count)
        out_heatmap.write(overlay_frame)  # For heatmap video (with heatmap overlay)
        out_distancing.write(frame)  # Write distancing frame regardless of violation for consistency

    # Release resources
    cap.release()
    out_detection.release()
    out_heatmap.release()
    out_distancing.release()  # Release the distancing output video
    cv2.destroyAllWindows()
```

```
/usr/local/lib/python3.10/dist-packages/torch/hub.py:330: UserWarning: You are about to download and run code from an untrusted repository. In a f
  warnings.warn(
Downloading: "https://github.com/ultralytics/yolov5/zipball/master" to /root/.cache/torch/hub/master.zip
Creating new Ultralytics Settings v0.0.6 file ✅
View Ultralytics Settings with 'yolo settings' or at '/root/.config/Ultralytics/settings.json'
Update Settings with 'yolo settings key=value', i.e. 'yolo settings runs_dir=path/to/dir'. For help see https://docs.ultralytics.com/quickstart/#u
YOLOv5 🚀 2024-11-11 Python-3.10.12 torch-2.5.0+cu121 CPU

Downloading https://github.com/ultralytics/yolov5/releases/download/v7.0/yolov5x.pt to yolov5x.pt...
100%|██████████| 166M/166M [00:00<00:00, 182MB/s]

Fusing layers...
```