A

Project Report

on

MobCurrent: YOLO-Based People Flow Detection and Interactive Visual Analytics

Submitted in partial fulfillment of the requirement

For the award of the degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

(ARTIFICIAL INTELLIGENCE)

Submitted by

Lanka Lakshmi Prasanna

(21A31A4314)

Under the esteemed supervision of

Mrs. T. Tejasvi , M.Tech, (Ph.D).,

Assistant Professor of CSE (AI).

DEPARTMENT OF CSE (ARTIFICIAL INTELLIGENCE)

PRAGATI ENGINEERING COLLEGE

(AUTONOMOUS)

(Approved by AICTE, Permanently Affiliated to JNTUK, KAKINADA, Accredited by NBA & NAAC with 'A+' Grade)

ADB Road, Surampalem, Near Peddapuram, Kakinada District, AP- 533437

2021-2025

MobCurrent: YOLO-Based People Flow Detection and Interactive Visual Analytics

A

Project Report

on

MobCurrent: YOLO-Based People Flow Detection and Interactive Visual Analytics

Submitted in partial fulfillment of the requirement

For the award of the degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

(ARTIFICIAL INTELLIGENCE)

Submitted by

Lanka Lakshmi Prasanna

(21A31A4314)

Under the esteemed supervision of

Mrs. T. Tejasvi , M.Tech, (Ph.D).,

Assistant Professor of CSE (AI).

DEPARTMENT OF CSE (ARTIFICIAL INTELLIGENCE)

PRAGATI ENGINEERING COLLEGE

(AUTONOMOUS)

(Approved by AICTE, Permanently Affiliated to JNTUK, KAKINADA, Accredited by NBA & NAAC with ‘A+’ Grade)

ADB Road, Surampalem, Near Peddapuram, Kakinada District, AP- 533437

2021-2025

PRAGATI ENGINEERING COLLEGE

(AUTONOMOUS)

(Approved by AICTE, Permanently Affiliated to JNTUK, Kakinada, Accredited by NBA & NAAC with ‘A+’ Grade)

ADB Road, Surampalem, Near Peddapuram, Kakinada District, AP-533437

CERTIFICATE

DEPARTMENT OF CSE (ARTIFICIAL INTELLIGENCE)

Learning is Supreme Deity

This is to certify that the project report entitled **“MobCurrent: YOLO-Based People Flow Detection and Interactive Visual Analytics”** is being submitted by **Lanka Lakshmi Prasanna (**21A31A4314**)** in partial fulfilment for the award of the Degree of **Bachelor of Technology**, during the year **2021-2025** in CSE (ARTIFICIAL INTELLIGENCE) of Pragati Engineering College, for the record of a bonified work carried out by her.

Project Supervisor:

Mrs. T. Tejasvi ,M.Tech, (Ph.D).,
Assistant Professor
Department of CSE (AI)
Pragati Engineering College (A)

Head of the Department:

Mrs. K. Lakshmi Viveka,M.Tech, (Ph.D).,
Associate Professor & HOD
Department of CSE (AI)
Pragati Engineering College (A)

Project Co-Supervisor:

Mrs. K. Lakshmi Viveka,M.Tech,(Ph.D).,
Associate Professor & HOD
Department of CSE (AI)
Pragati Engineering College (A)

External Signature

ACKNOWLEDGEMENTS

I express our thanks to project supervisor **Mrs. T. Tejasvi**, **Assistant Professor of Computer Science and Engineering (Artificial Intelligence),** who deserves a special note of thanks and gratitude, for having extended their fullest cooperation and guidance, without this, the project would never have materialized.

I express our deep sense of gratitude to **Mrs. K. Lakshmi Viveka**, **Associate Professor and Head of the Department of Computer Science and Engineering (Artificial Intelligence)**, for having shown keen interest at every stage of development of my project and for guiding me in every aspect.

I wish to express my special thanks to our beloved **Dr. K. SATYANARAYANA, Professor & Director (Academics)** for giving guidelines and encouragement. And i also wish to express my special thanks to our beloved **Dr. G. NARESH, Professor & Principal** for giving guidelines and encouragement.

I wish to express sincere gratitude to our beloved and respected **Sri. M. SATISH, Vice-President**, **Sri. M. V. HARANATHA BABU, Director (Management)** and **Dr. P. KRISHNA RAO, Chairman** for their encouragement and blessings.

I am thankful to all our faculty members of the Department for their valuable suggestions. My sincere thanks are also extended to all the teaching and non-teaching staff of Pragati Engineering College.

LANKA LAKSHMI PRASANNA
(21A31A4314)

ABSTRACT

MobCurrent is a real-time people flow detection system leveraging YOLO (You Only Look Once) and DeepSORT for object detection and tracking. The system integrates a Flask-based web interface for interactive visual analytics, allowing users to upload and process videos to detect unique individuals within a camera's range. The processed data is stored and visualized using Matplotlib and Plotly for insightful analytics. This document explores the motivation, problem statement, and scope of the project, along with a literature survey highlighting advancements in object detection and tracking techniques. The growing need for automated people flow detection in crowded environments necessitates an efficient, scalable, and accurate solution. By employing deep learning models, MobCurrent enhances detection accuracy, reduces redundant tracking errors, and provides users with meaningful insights. This solution finds applications in retail analytics, security surveillance, and urban infrastructure planning

MobCurrent is a real-time people flow detection system leveraging YOLO (You Only Look Once) and DeepSORT for object detection and tracking. The system integrates a Flask-based web interface for interactive visual analytics, allowing users to upload and process videos to detect unique individuals within a camera's range. The processed data is stored and visualized using Matplotlib and Plotly for insightful analytics. This document explores the motivation, problem statement, and scope of the project, along with a literature survey highlighting advancements in object detection and tracking techniques.

The growing need for automated people flow detection in crowded environments necessitates an efficient, scalable, and accurate solution. Traditional surveillance systems often struggle with issues such as overlapping objects, motion blur, and poor lighting conditions. MobCurrent addresses these challenges by employing advanced deep learning models that enhance detection accuracy, reduce redundant tracking errors, and provide meaningful insights.

As cities and industries become more data-driven, systems like MobCurrent play a critical role in bridging the gap between digital intelligence and real-world decision-making, enabling smarter, more adaptive environments. Ultimately, it leads to safer, more accessible, and sustainable urban environments.

TABLE OF CONTENTS

LIST OF FIGURES & TABLES

CHAPTER-1

INTRODUCTION

INTRODUCTION

MobCurrent is an advanced visual analytics tool designed to monitor human movement in crowded environments. Traditional methods for crowd analysis lack accuracy and efficiency in real-time scenarios. This project integrates YOLO and DeepSORT to provide a seamless experience for tracking unique individuals within a video feed. The system employs Flask for web-based interaction, offering a lightweight yet robust framework that ensures accessibility, ease of deployment, and seamless user experience across various devices and platforms.

People flow detection plays a crucial role in various domains, including security surveillance, retail analytics, and urban planning. Existing solutions often struggle with re-identification issues, leading to inaccurate headcounts. By leveraging YOLO for real-time object detection and DeepSORT for identity preservation, MobCurrent enhances accuracy and reliability. The system provides an intuitive web-based interface where users can upload video footage for processing.

The importance of accurate people flow detection has grown significantly in recent years, especially in environments such as airports, shopping malls, and transportation hubs. Effective crowd management in such spaces requires precise tracking of individuals to prevent congestion, enhance security, and optimize resource allocation. MobCurrent addresses these challenges by combining state-of-the-art deep learning models with a user-friendly interface.

The Flask web interface plays a pivotal role in MobCurrent's usability. By offering a simple yet powerful platform for uploading video files, users can easily interact with the system without extensive technical expertise. The interface presents processed data in a clear and informative manner using visual tools like Matplotlib for static charts and Plotly for interactive data exploration. This feature ensures that decision-makers can analyze trends, identify peak activity periods, and make informed decisions to improve crowd management strategies.

This document provides a comprehensive analysis of the system's purpose, scope, motivation, and problem statement. The rapid advancements in AI, deep learning models have enabled real-time analytics for people tracking, making this system a crucial tool for decision-makers. By continuously monitoring environments such as airports, shopping malls, stadiums, and public transport hubs, these systems can detect crowd density, individual movement patterns, and unusual behaviors with high accuracy.

1.1 Purpose

The primary objective of MobCurrent is to develop a robust, real-time people flow detection system that ensures high accuracy in identifying unique individuals within a monitored space. Conventional systems often fail to differentiate between individuals re-entering the frame, leading to inflated counts. MobCurrent addresses this issue by integrating advanced tracking algorithms with object detection models.

Additionally, the system aims to facilitate data-driven decision-making through interactive analytics. By providing a seamless web interface, users can visualize daily person counts and gain insights into peak hours, crowd density, and movement trends. This enhances applications in security, retail management, and event planning.

To ensure optimal performance across diverse environments, MobCurrent is designed with flexibility in mind. The system leverages adaptive detection thresholds, ensuring consistent accuracy even in environments with fluctuating lighting conditions, partial occlusions, or dense crowd formations. This adaptability enhances MobCurrent's reliability in both indoor and outdoor spaces, ensuring it meets the needs of various industries and institutions.

The project also aims to reduce hardware dependency by ensuring that the system runs on standard computing devices without requiring high-end GPUs. Through efficient model optimization, MobCurrent maintains fast inference speeds with minimal resource consumption. This makes it feasible for small businesses and public institutions to adopt people tracking solutions at a lower cost. Additionally, MobCurrent can be deployed on cloud platforms, further reducing hardware costs while ensuring scalability.

Furthermore, MobCurrent ensures that all data is stored securely, maintaining privacy standards while offering detailed analytics. The system employs encryption techniques for data storage and transfer, ensuring that sensitive information is protected from unauthorized access. This focus on data security aligns with industry best practices, making MobCurrent a trusted solution for organizations handling large volumes of personal data.

By combining high accuracy, scalability, cost-effectiveness, and secure data handling, MobCurrent provides a comprehensive solution for modern crowd analysis and monitoring challenges. Its flexible design and user-friendly interface empower businesses, security agencies, and public institutions to make informed decisions based on real-time insights.

1.2 Scope

MobCurrent is designed to operate in dynamic environments such as shopping malls, transportation hubs, and public events. The system processes video inputs and tracks human movement without requiring specialized hardware. It supports multiple formats and ensures scalability for real-time applications. The project encompasses object detection, tracking, and visualization. YOLO handles fast and accurate detection, while DeepSORT ensures individual re-identification across frames.

The system is versatile enough to cater to diverse sectors such as healthcare facilities, educational institutions, and retail outlets. For example, hospitals can use MobCurrent to monitor patient flow, ensuring compliance with safety protocols and minimizing overcrowded spaces. This helps optimize resource allocation, improve response times in critical areas, and enhance overall patient experience and operational efficiency.

MobCurrent's architecture is designed to accommodate future enhancements that improve functionality and expand its use cases. Planned upgrades include multi-camera support, which will enable simultaneous monitoring of multiple entry points, improving security in large public spaces. Additionally, AI-driven predictive analysis is intended to enhance the system's ability to forecast crowd behavior patterns. This proactive approach enables early interventions, helping prevent congestion and ensure smoother crowd flow in real-time.

Cloud integration is another key development focus for MobCurrent. By leveraging cloud computing, the system will achieve improved scalability, ensuring seamless performance even when processing large volumes of video data.

The system’s adaptability makes it a valuable tool for both security agencies and businesses looking to optimize space utilization. MobCurrent's compatibility with existing surveillance infrastructure ensures that organizations can adopt it without extensive hardware replacements. This seamless integration not only reduces implementation costs but also accelerates deployment timelines, allowing businesses and security agencies to enhance their monitoring capabilities with minimal disruption.

With machine learning-based predictive analytics, the system can also forecast crowd trends and suggest optimal movement strategies to avoid congestion. This feature will empower businesses, event organizers, and security personnel to make data-driven decisions that enhance safety, improve customer experience, and maximize operational efficiency.

1.3 Motivation

The increasing need for real-time crowd monitoring has driven the development of MobCurrent. Traditional manual counting methods are labor-intensive and error-prone, while conventional surveillance lacks intelligent analytics. By automating the process, the system minimizes human intervention and enhances accuracy. Another driving factor is the potential application of people flow analytics in optimizing urban infrastructure.

The fusion of YOLO and DeepSORT bridges the gap between efficiency and precision in people tracking. YOLO's rapid object detection combined with DeepSORT's robust re-identification mechanism ensures individuals are tracked consistently across multiple frames. This combination significantly reduces false positives, making the system reliable in busy and complex environments. MobCurrent's ability to operate efficiently in high-traffic locations such as airports, train stations, and entertainment venues further emphasizes its relevance.

The motivation behind MobCurrent also stems from the advancements in deep learning and computer vision, which have made high-performance object detection and tracking feasible even on consumer-grade hardware.

Furthermore, MobCurrent's automated tracking system reduces the cognitive load on security personnel by eliminating redundant detections. This ensures security teams can focus on real threats instead of dealing with false alarms. By streamlining crowd monitoring, MobCurrent empowers decision-makers with actionable insights, improving operational efficiency across various domains. Whether in transportation hubs, public venues, retail spaces, or large-scale events, the system provides real-time data that supports timely interventions, resource optimization, and enhanced situational awareness.

The motivation to develop MobCurrent also considers the growing demand for data-driven decision-making. In retail environments, understanding customer movement patterns enables businesses to optimize store layouts, improve product placement, and enhance customer engagement strategies.

By combining high-performance object detection, real-time tracking, and actionable insights, MobCurrent addresses the limitations of traditional monitoring systems while offering a scalable solution for future innovations in AI-driven crowd analytics.

1.4 Problem Statement

Current people flow detection systems face challenges such as identity re-identification, real-time processing, and scalability. Traditional surveillance methods lack intelligent tracking, leading to inaccurate estimations of unique individuals. Moreover, existing solutions often require expensive hardware setups, limiting accessibility. MobCurrent aims to resolve these challenges by implementing an efficient, AI-powered approach to detect and track unique persons in crowded areas. The system ensures minimal computational overhead while maintaining high detection accuracy. By integrating a user-friendly web interface, it bridges the gap between technical complexity and practical usability.

One of the primary issues in traditional tracking systems is their inability to handle re-identification efficiently. When individuals exit and re-enter the frame, many conventional solutions struggle to distinguish them as the same person, resulting in duplicated counts. MobCurrent addresses this through DeepSORT's robust re-identification mechanism, ensuring consistent identity tracking across frames. This feature significantly improves data accuracy, enabling businesses and security agencies to make reliable decisions based on precise crowd analytics. With enhanced precision in detecting movement patterns and density fluctuations, MobCurrent helps reduce false positives and ensures that responses are based on real-world conditions.

Real-time processing is another critical challenge that MobCurrent overcomes. Leveraging YOLO's fast object detection capabilities, the system processes video streams with minimal latency, ensuring timely insights for decision-makers. This feature is especially crucial for scenarios requiring instant response, such as security incidents, crowd control, and emergency evacuations. By employing efficient deep learning models, MobCurrent achieves this without relying on expensive hardware, making it suitable for budget-conscious organizations.

Scalability is a key aspect of MobCurrent's design. The system can handle multiple video feeds simultaneously, making it viable for large-scale deployments such as stadiums, train stations, or shopping malls. This capability allows organizations to centralize their surveillance and analytics efforts, reducing operational costs and improving overall efficiency.

In addition to detection and tracking, MobCurrent addresses the challenge of presenting data in an actionable format. The system's intuitive visual analytics interface enables stakeholders to explore trends, analyze peak activity periods, and assess crowd movement patterns.

CHAPTER-2

LITERATURE SURVEY

LITERATURE SURVEY

People flow detection and tracking have evolved significantly with advancements in deep learning and computer vision. Earlier methods relied on manual counting or simple motion detection techniques, which often resulted in inaccuracies. With the advent of deep learning, object detection models like YOLO and Faster R-CNN have drastically improved accuracy and speed. YOLO has gained popularity due to its real-time detection capabilities. Unlike traditional sliding window methods, YOLO treats object detection as a single regression problem, making it highly efficient. However, object detection alone is insufficient for tracking individuals across multiple frames. To address this, tracking algorithms like DeepSORT have been introduced, ensuring identity preservation across video sequences. Recent studies have demonstrated the effectiveness of combining YOLO with DeepSORT for people tracking.

The development of DeepSORT has been crucial in enhancing people tracking accuracy. DeepSORT employs a combination of motion information and appearance features to maintain identity consistency even in crowded and complex environments. This approach significantly improves re-identification accuracy by distinguishing between individuals with similar appearances, reducing duplicate counts and false positives. By incorporating advanced feature extraction techniques and leveraging spatial-temporal context, the system ensures consistent tracking across multiple frames and camera views.

Research has also explored hybrid techniques that combine object detection with temporal data analysis. For instance, techniques leveraging Long Short-Term Memory (LSTM) networks have been shown to improve trajectory prediction and movement pattern analysis. Such methods enhance the system's ability to predict crowd dynamics, which is valuable in scenarios requiring proactive decision-making, such as emergency evacuations or event management.

In addition to model improvements, advancements in hardware acceleration have played a significant role in enhancing real-time processing capabilities. With the availability of consumer-grade GPUs and optimized inference engines like TensorRT and ONNX Runtime, deploying complex deep learning models on edge devices has become feasible. This shift has enabled systems like MobCurrent to deliver accurate results without relying on expensive, specialized hardware, making advanced crowd analytics more accessible, cost-effective, and scalable for a wide range of industries.

CHAPTER-3

SYSTEM ANALYSIS

SYSTEM ANALYSIS

3.1 Introduction

System analysis plays a crucial role in understanding the requirements, challenges, and technical considerations associated with the development of MobCurrent. The project aims to provide a robust and scalable solution for real-time people flow detection using YOLO and DeepSORT algorithms. The system integrates a web-based interface developed using Flask, allowing users to upload and process videos to detect unique individuals. The processed data is stored and visualized through Matplotlib and Plotly, enabling interactive analysis of people flow trends.

In this section, we examine the existing system's limitations, the improvements offered by the proposed system, and the feasibility of implementing MobCurrent in real-world applications. The system's ability to accurately track and count unique individuals is essential for various applications, including security surveillance, crowd management, and retail analytics.

The limitations of traditional people tracking systems are evident in their reliance on simple motion detection algorithms or manual counting methods. These approaches are prone to errors, especially in crowded environments where overlapping objects, occlusions, and rapid movement can hinder accurate counting. Additionally, many conventional systems struggle with identity re-identification, often leading to inflated counts or misinterpretations of movement patterns. Such limitations compromise the reliability of data, making it difficult for decision-makers to extract meaningful insights.

MobCurrent addresses these issues by combining YOLO's powerful object detection capabilities with DeepSORT's advanced tracking mechanisms. YOLO's ability to detect multiple objects in a single frame ensures fast and efficient identification of individuals. DeepSORT enhances this by tracking each detected individual across multiple frames using motion prediction and appearance-based features. This synergy reduces false positives, improves identity preservation, and ensures accurate people flow tracking. By harmonizing visual recognition with contextual reasoning, the system maintains continuity in tracking individuals even during occlusions or sudden movements, thereby enhancing overall reliability and precision in dynamic, real-world environments. This integrated approach ensures that people flow analytics remain consistent and trustworthy and even in complex scenarios with frequent visual disruptions or in the high crowd density.

3.2 Existing System

Traditional people flow detection systems rely on manual counting, simple motion detection algorithms, or basic object detection techniques. Many surveillance systems utilize fixed-position cameras with predefined thresholds for detecting movement. These systems often fail to differentiate between unique individuals, leading to inaccurate counts. Moreover, existing tracking methods struggle with occlusion, re-identification, and varying lighting conditions, resulting in unreliable data.

Some systems use background subtraction techniques to detect motion; however, these methods are prone to errors in dynamic environments where lighting conditions and camera angles change frequently. Background subtraction methods often fail in crowded areas, where the presence of multiple overlapping individuals creates visual noise. Such systems also struggle to maintain identity tracking when individuals leave and re-enter the camera frame, resulting in duplicate counting or loss of tracking continuity.

While some AI-powered solutions have been introduced to address these challenges, they often come with significant drawbacks. Many AI-based systems demand powerful hardware resources, such as GPUs or dedicated servers, which can be costly for small businesses or organizations with limited budgets. Additionally, some existing solutions require extensive model training and customization to adapt to different environments, making deployment complex and resource-intensive.

Furthermore, scalability remains a major concern in traditional systems. Expanding surveillance coverage often requires additional hardware investments and infrastructure modifications. These limitations hinder the widespread adoption of traditional systems in dynamic, high-traffic environments like shopping malls, train stations, and public events where real-time accuracy is crucial.

Overall, existing people flow detection systems face critical challenges in accuracy, scalability, and deployment complexity. These limitations emphasize the need for an efficient, lightweight, and user-friendly solution that addresses these shortcomings while ensuring reliable performance in diverse environments. MobCurrent aims to fill this gap by providing an AI-driven approach that combines robust detection models with advanced tracking techniques, improving accuracy, efficiency, and scalability for real-world applications.

3.2.1 Drawbacks

The primary drawbacks of existing people flow detection systems include:

- **Inaccurate Tracking:** Many systems fail to re-identify individuals correctly, leading to over-counting or under-counting errors.
- **Lack of Real-Time Processing:** Traditional methods rely on batch processing rather than real-time analysis, causing delays in decision-making.
- **High Hardware Requirements:** Advanced AI-based solutions often require expensive GPUs and high-performance computing infrastructure.
- **Limited Data Visualization:** Most existing systems lack interactive analytics, making it difficult for users to interpret trends and insights effectively.
- **Scalability Issues:** Current methods are not optimized for handling multiple video feeds or large-scale surveillance applications.

These limitations highlight the need for an improved system that can provide accurate real-time tracking, scalable infrastructure, and meaningful data visualization, which MobCurrent aims to achieve.

Traditional crowd monitoring systems often suffer from inconsistent object detection, occlusions, and duplicate counting, leading to inaccurate results. By leveraging YOLO for robust real-time object detection and DeepSORT for unique identity tracking, MobCurrent significantly reduces false positives and ensures precise person re-identification across video frames. This improved tracking mechanism enhances movement pattern analysis, anomaly detection, and congestion monitoring, making it an ideal solution for large-scale public spaces and events. It also supports real-time decision-making by providing actionable insights to manage crowds efficiently and ensure public safety.

Scalability is a major challenge in real-time video analytics, as large volumes of video data require efficient processing and storage. Traditional systems often struggle with high latency, slow inference speeds, and limitations in handling multiple video streams. MobCurrent addresses these issues by integrating GPU-accelerated deep learning models, cloud-based deployment options, and containerized microservices. By using Kubernetes for orchestration and distributed computing frameworks, the system ensures that it can seamlessly scale to accommodate increasing workloads, whether deployed on-premises or in the cloud.

3.3 Proposed System

MobCurrent introduces an AI-driven solution that overcomes the drawbacks of traditional people flow detection systems. It integrates YOLO for real-time object detection and DeepSORT for efficient tracking, ensuring precise identification of unique individuals within a monitored space. The system is designed to operate on standard computing devices, reducing dependency on expensive hardware.

MobCurrent's architecture includes a Flask-based web interface that allows users to upload video footage for processing. The processed results are displayed through interactive dashboards, leveraging Matplotlib and Plotly for data visualization. This web interface provides a user-friendly experience, ensuring accessibility for individuals without technical expertise. Users can analyze crowd flow trends, identify peak activity hours, and extract meaningful insights for better decision-making. The system's intuitive design reduces the complexity typically associated with AI-based tracking solutions. This user-friendly interface allows for quick adoption and seamless integration into existing monitoring infrastructures.

MobCurrent also emphasizes scalability. By utilizing lightweight models and efficient tracking algorithms, the system can handle multiple video feeds simultaneously without excessive computational load. This makes it suitable for large-scale deployments in environments like shopping malls, stadiums, or transportation hubs. Additionally, the modular design of MobCurrent enables easy integration with existing surveillance infrastructure, reducing deployment costs for organizations.

To address security concerns, MobCurrent includes secure data storage mechanisms that comply with privacy regulations. The system ensures that no personally identifiable information is stored, focusing only on anonymized crowd movement data. This feature makes MobCurrent a viable solution for applications in public spaces where privacy concerns are paramount.

With its combination of advanced detection algorithms, efficient tracking, and user-friendly analytics, MobCurrent presents a comprehensive solution for real-time people flow detection. Its lightweight yet powerful architecture empowers businesses, event organizers, and security agencies to make data-driven decisions, improving crowd management strategies and ensuring public safety. With seamless integration and minimal computational overhead, it enables real-time deployment in diverse environments, from busy transit hubs to large-scale public gatherings, enhancing responsiveness and situational awareness.

3.3.1 Advantages

The proposed system offers several key advantages over traditional people flow detection methods:

- ❖ **Real-Time Processing:** The system detects and tracks individuals in real time, providing instant insights for decision-making. This enables security personnel, event organizers, and retail managers to respond swiftly to crowd movements, improving safety and customer experience.
- ❖ **Improved Accuracy:** By combining YOLO with DeepSORT, MobCurrent minimizes re-identification errors and ensures accurate person counting. DeepSORT's identity preservation mechanism effectively distinguishes between individuals even in crowded and fast-moving environments.
- ❖ **Scalability:** The system can be deployed across multiple locations without requiring extensive modifications. Its lightweight architecture allows organizations to scale their monitoring infrastructure as needed, making it ideal for large facilities like airports, train stations, and shopping centers.
- ❖ **User-Friendly Interface:** The web-based application allows users to easily upload, process, and analyze videos without requiring technical expertise. The interface is designed with simplicity in mind, ensuring that stakeholders can access actionable insights without the need for extensive training.
- ❖ **Cost-Effective:** Unlike high-end AI solutions, MobCurrent is designed to run on consumer-grade hardware, making it accessible for businesses of all sizes. This cost-efficiency lowers entry barriers for small enterprises, public institutions, and educational facilities looking to implement people flow analytics.
- ❖ **Comprehensive Data Visualization:** Interactive charts and graphs help users interpret people flow patterns efficiently. Using Matplotlib for static charts and Plotly for interactive visuals, the system presents insights in an intuitive manner. This visualization approach empowers decision-makers with actionable information, enabling them to plan staffing, security protocols, or promotional activities effectively.

By offering a powerful blend of accuracy, flexibility, and usability, MobCurrent addresses the limitations of traditional surveillance systems while introducing a scalable and cost-effective solution for diverse industries. Its modular design supports easy customization, allowing organizations to adapt the system to specific operational needs, whether it's managing retail foot traffic, enhancing event security, or monitoring public infrastructure.

3.4 Feasibility Study

Before deploying MobCurrent, it is essential to evaluate its feasibility across economic, operational, and technical aspects. This study ensures that the system is practical, cost-effective, and capable of fulfilling its intended objectives.

3.4.1 Economic Feasibility

Economic feasibility assesses whether the project is financially viable and provides a return on investment. MobCurrent reduces operational costs by minimizing the need for manual monitoring and expensive surveillance infrastructure. Traditional surveillance systems require high-end servers and dedicated personnel to monitor and analyze data, whereas MobCurrent automates these processes using AI-powered tracking. This automation not only reduces operational costs but also enhances accuracy and response time in dynamic environments.

The system's reliance on open-source frameworks such as Flask, TensorFlow, and YOLO further reduces software licensing costs. Businesses and organizations can deploy MobCurrent without significant financial investment, making it a cost-effective solution for real-time people flow detection. Additionally, the implementation of cloud-based processing can further optimize costs, allowing organizations to scale based on demand without maintaining expensive on-premises hardware.

The system’s efficient use of computing resources further contributes to cost savings. MobCurrent processes video data in real-time without overloading system memory or CPU, ensuring that organizations can implement the solution without expensive hardware upgrades. This makes it feasible for budget-conscious organizations seeking to improve their crowd monitoring capabilities without extensive financial investment. By leveraging scalable technology, MobCurrent ensures accessibility without compromising on performance or reliability.

In addition, MobCurrent’s modular architecture allows organizations to implement features incrementally, reducing upfront costs. Businesses can start with basic tracking functionalities and expand to advanced analytics and predictive insights as their needs grow. This flexible implementation approach enhances the system’s overall cost-effectiveness, making MobCurrent an ideal solution for organizations of all sizes looking to improve operational efficiency through intelligent people flow detection.

3.4.2 Operational Feasibility

Operational feasibility determines whether the system can be effectively implemented and used in real-world scenarios. MobCurrent is designed to be user-friendly, with an intuitive interface that allows users to upload videos and access analytical insights effortlessly. The system can be integrated with existing surveillance cameras, enabling seamless adoption without disrupting ongoing operations.

One of the major operational benefits is the automation of people flow analysis. Security personnel, event managers, and business owners can leverage MobCurrent to monitor real-time crowd movement without requiring technical expertise. The system also ensures data privacy by processing videos locally, preventing unauthorized access to sensitive information. The ability to generate reports and visualize trends enhances its practical usability across different industries. These insights support long-term planning, performance evaluation, and the continuous improvement of operational strategies.

MobCurrent's adaptability to diverse environments makes it ideal for deployment in public spaces such as malls, airports, stadiums, and educational institutions. For instance, in retail settings, managers can analyze customer movement patterns to identify busy sections within the store, optimizing product placement and staffing strategies that visualize trends enhances its practical usability across different industries.. In transportation hubs, administrators can utilize the system to monitor passenger flow and reduce congestion during peak hours. Additionally, MobCurrent's capability to track movement in large venues like concerts and sports arenas allows organizers to manage crowd density and improve security arrangements effectively.

Furthermore, MobCurrent offers customizable alert systems that notify users when abnormal crowd behavior or excessive congestion is detected. This proactive feature enhances safety measures in high-traffic areas. The system's scalability ensures that it can handle multiple camera feeds simultaneously, making it suitable for organizations of various sizes. With its automated workflow and interactive analytics, MobCurrent streamlines operational processes, reduces manual intervention, and improves overall decision-making. By providing actionable insights in an accessible format, MobCurrent empowers users to enhance crowd management strategies while minimizing resource expenditure. Moreover, its intuitive interface and real-time analytics allow stakeholders to proactively respond to crowd dynamics, identify bottlenecks, and implement timely interventions, ultimately fostering operational excellence across diverse scenarios.

3.4.3 Technical Feasibility

Technical feasibility examines whether the system can be developed and deployed using the available technology. MobCurrent is built using widely adopted AI frameworks, making it highly compatible with existing hardware and software environments. The system is designed to run on standard computing devices without requiring dedicated AI accelerators or high-end GPUs. This design choice reduces hardware costs and allows for broader deployment in resource-constrained environments such as small businesses, educational institutions, and public facilities.

The combination of YOLO and DeepSORT ensures efficient object detection and tracking, optimizing processing speed without compromising accuracy. YOLO's one-stage detection approach enhances real-time performance, while DeepSORT's robust re-identification mechanism maintains individual identities across multiple frames. These features make MobCurrent capable of handling crowded environments with frequent movement and occlusions.

The use of Flask for the web interface allows for seamless deployment across multiple platforms. Flask's lightweight architecture ensures that the system can operate on low-specification devices, while its flexibility allows developers to customize the interface to suit specific business requirements. Additionally, the integration of data visualization tools like Matplotlib and Plotly enhances the system's analytical capabilities.

The modular design of MobCurrent makes it easy to extend and customize for future enhancements. The architecture is structured to accommodate additional features such as multi-camera support, cloud-based analytics, and AI-driven predictive modeling. By implementing cloud integration, MobCurrent can handle large-scale video data processing, ensuring scalability for enterprise environments. Moreover, developers can incorporate additional AI models to improve tracking accuracy in challenging conditions such as low light, heavy occlusions, or rapidly changing environments. This flexibility ensures MobCurrent remains robust and effective even in complex and unpredictable real-world scenarios.

By addressing economic, operational, and technical feasibility, MobCurrent proves to be a scalable, cost-effective, and practical solution for people flow detection and interactive visual analytics. Its adaptability across diverse sectors and the ability to function with minimal infrastructure make it a powerful tool for enhancing security, optimizing business operations, and improving crowd management strategies.

3.4.4 Social Feasibility

Social feasibility examines the acceptance and impact of MobCurrent within society. As a real-time people flow monitoring and analysis platform, MobCurrent is designed to enhance safety, optimize crowd management, and support decision-making across various public and private sectors. The key social feasibility aspects include:

Enhancing Public Safety: By monitoring crowd density and detecting unusual movement patterns, MobCurrent can aid in early detection of emergencies, supporting rapid response and ensuring public safety in crowded areas like malls, airports, and event venues.

Accessibility and Inclusion: MobCurrent's visual dashboards and real-time alerts provide inclusive access for operators with different needs, allowing seamless interpretation of crowd data without the need for complex technical know-how.

Educational Support: The system can be used in educational institutions to study human behavior, crowd dynamics, and the use of AI in real-world monitoring systems, thereby offering a valuable learning tool for students and researchers.

3.4.5 Legal Feasibility

Legal feasibility plays a crucial role in the deployment of MobCurrent. The system processes sensitive video data and requires compliance with various legal standards related to surveillance, privacy, and data protection. The key legal considerations include:

Compliance with Data Protection Laws: As MobCurrent collects and processes visual data of individuals in public and semi-public spaces, it must comply with global data protection frameworks such as the General Data Protection Regulation (GDPR) and India's Digital Personal Data Protection Act, ensuring data is anonymized, encrypted, and securely handled.

Usage of Surveillance Equipment: The system must follow legal protocols for video surveillance in public areas, including consent (where required), proper signage, and restriction of camera placement to avoid infringing personal privacy.

Data Retention and Anonymization: MobCurrent should implement policies for minimal data retention and must anonymize identifiable information unless explicitly authorized, to stay aligned with ethical AI practices and legal mandates.

CHAPTER-4

SYSTEM DESIGN AND IMPLEMENTATION

SYSTEM DESIGN AND IMPLEMENTATION

4.1 System Architecture

MobCurrent's architecture is designed to integrate multiple components that efficiently handle video processing, data visualization, and user interaction. The modular design ensures scalability, flexibility, and seamless functionality.

Core Components of the System Architecture

The system architecture comprises the following key components:

- **Camera/Input Source**
  - The camera or input source is responsible for capturing video data. This data can come from live surveillance feeds or pre-recorded video files.
  - For convenience, MobCurrent's web interface allows users to upload these video files directly. This flexibility ensures compatibility with a variety of video sources, including CCTV systems, mobile devices, and drones.
  - The system supports common video formats such as .mp4, .avi, and .mkv to enhance compatibility.
- **Processing Unit**
  - The processing unit is the core engine where AI-based detection and tracking are executed.
  - **YOLO (You Only Look Once)** is employed for real-time object detection, leveraging its speed and accuracy to identify individuals in each video frame.
  - **DeepSORT (Simple Online and Realtime Tracking)** is integrated to track individuals across multiple frames, ensuring accurate identification.
  - The processing unit is optimized to handle complex crowd environments, improving accuracy even in cases of occlusion, variable lighting, and dense populations.
  - The system employs GPU acceleration using CUDA-compatible devices to enhance performance for real-time processing.
- **Flask Web Application**
  - The Flask-based web interface acts as the user interaction layer.
  - Users can upload video files, initiate processing, and view analytical insights directly from this interface.

- ➢ The web interface offers intuitive navigation, ensuring accessibility for users with minimal technical expertise.
- ➢ Additionally, status indicators inform users about the progress of video uploads, processing completion, and error handling.

❖ **Storage System**

- ➢ Processed data is stored in structured formats for easy retrieval and analysis.
- ➢ The system initially adopts CSV files for simplicity but can be extended to SQL-based databases for improved scalability.
- ➢ The storage system logs information such as video metadata, timestamps, detected person counts, and movement trends.
- ➢ This structured data allows organizations to conduct long-term analytics and generate reports.

❖ **Data Visualization Module**

- ➢ The data visualization module leverages Matplotlib and Plotly to generate comprehensive visual insights.
- ➢ Graphs, bar charts, and heatmaps present user-friendly data interpretations, ensuring stakeholders can identify movement trends, peak crowd hours.
- ➢ The visual insights are dynamically updated to reflect new data, ensuring real-time monitoring capabilities.

❖ **User Dashboard**

- ➢ The dashboard consolidates all key insights into a centralized view, allowing users to monitor crowd patterns, analyze trends, and make informed decisions through an intuitive and interactive interface..
- ➢ Users can access processed video files, check detailed person counts, and analyze movement patterns directly from the dashboard.
- ➢ Features like filtering, sorting, and customizable visualizations enable tailored data exploration to suit different user requirements.

The components outlined above collectively form the backbone of the MobCurrent system architecture. Each module plays a vital role in ensuring the platform operates efficiently from capturing and processing video data to real-time object detection, tracking, and visual analytics. Together, these components work in harmony to deliver a scalable, responsive, and intelligent crowd monitoring solution that ensures accurate data processing, efficient system performance, and seamless user interaction.

4.2 UML Diagram

UML (Unified Modeling Language) diagrams help visualize the system's interactions and workflow. The primary UML diagrams for MobCurrent include the Use Case Diagram and the Sequence Diagram. These diagrams provide a clear understanding of system behavior, user roles, and the sequence of operations within different functional modules.

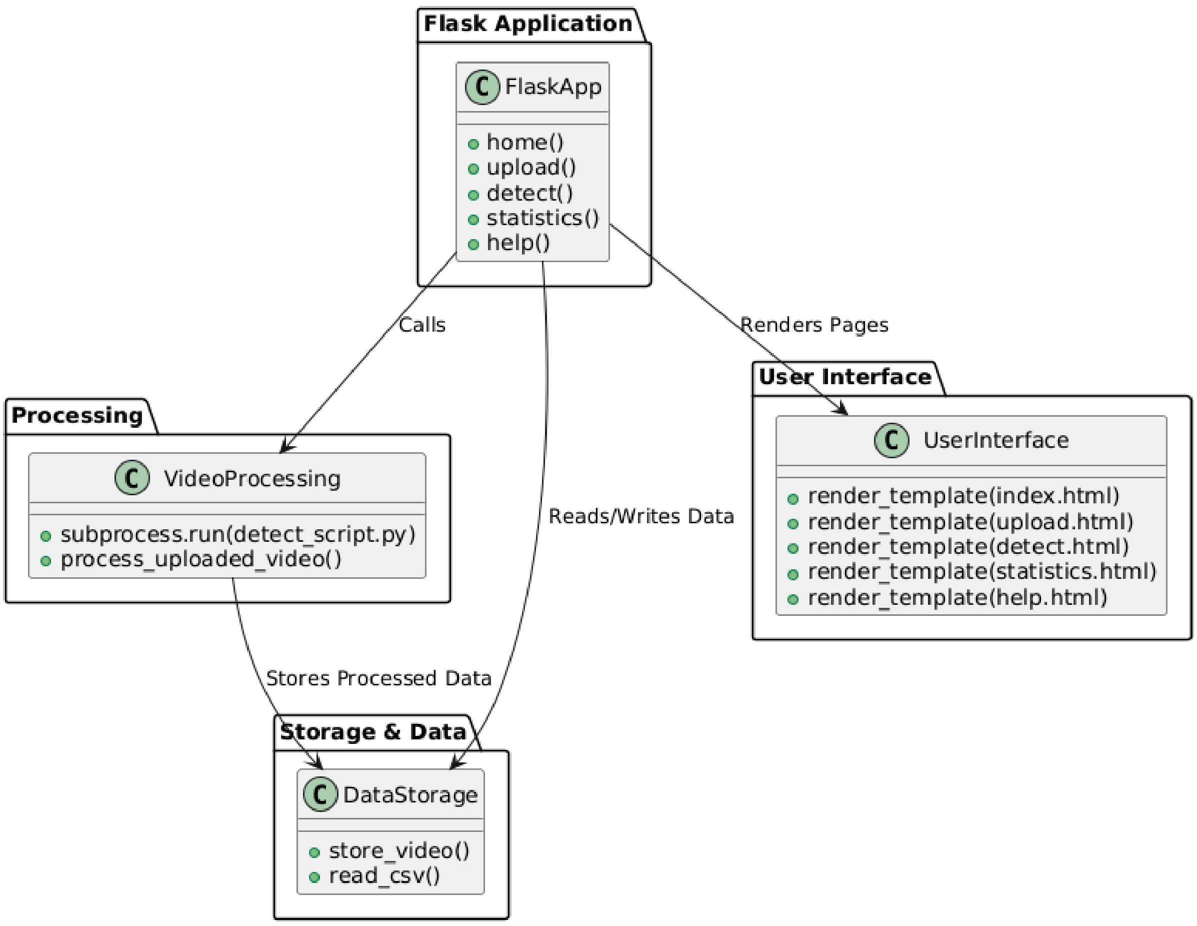


Figure 1: System Architecture Diagram

The architecture ensures efficient video processing, secure data handling, and seamless integration between various components. The modular design allows scalability for future enhancements, such as cloud-based processing and multi-camera support, enabling broader deployment and increased system flexibility.

4.2.1 Sequence Diagram

The following diagram represents the visual analytics of a bar graph generated using CSV data, structured in a format similar to a sequence diagram. It illustrates the flow of data from the CSV file to the final graphical representation. Initially, the system reads and processes the raw data from a CSV file, organizing it into structured columns such as timestamps and people count. This structured format allows for efficient data manipulation, enabling the system to extract meaningful insights and trends for visualization.

This processed data is then passed to the visualization module, where it is interpreted and mapped onto the bar graph. Each bar corresponds to a specific time segment or category, visually representing the count or trend observed during that period. The sequential layout of the diagram captures the step-by-step data transformation and rendering process, offering clear insights into how the raw data evolves into a meaningful visual summary for analytical and decision-making purposes.

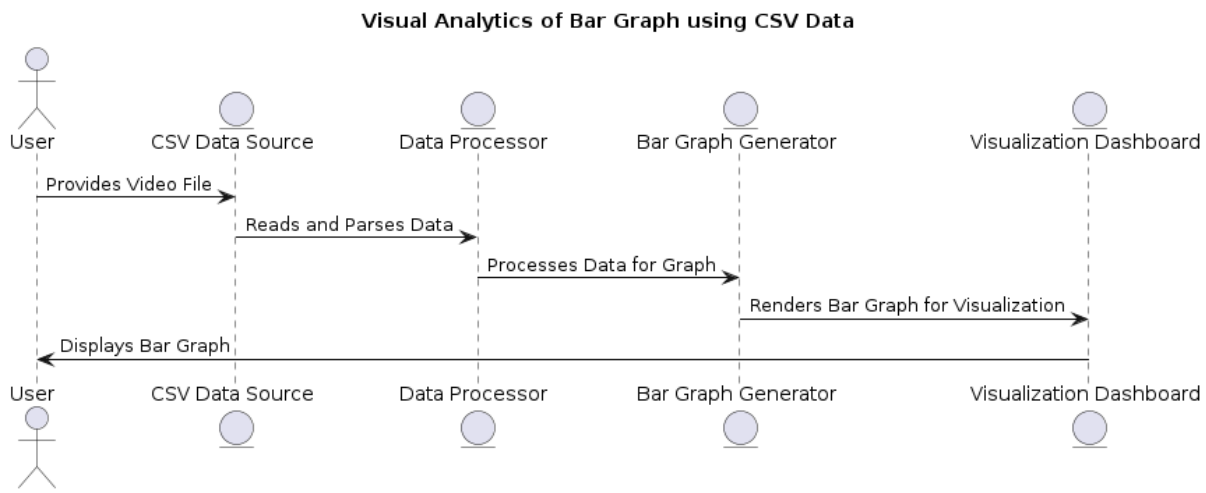


Figure 2: Visual Analytics of Bar Graph using CSV Data

The UML sequence diagram represents the workflow of processing a video file, extracting data into a CSV format, and using that data for visual analytics through a bar graph. It outlines each step in the system's interaction, from uploading the video, detecting and counting individuals, storing the results in a structured CSV file, and finally rendering the data into a bar graph that visually illustrates movement patterns or crowd flow over time.

4.2.2 Use case Diagram

The Use Case Diagram provides an overview of how users interact with the MobCurrent system. It highlights key user actions and system processes:

- **User Actions:** Upload Video, Process Video, View Analytics, Download Report.
- **System Processes:** YOLO Detection, DeepSORT Tracking, Database Storage, Web Application Interaction.

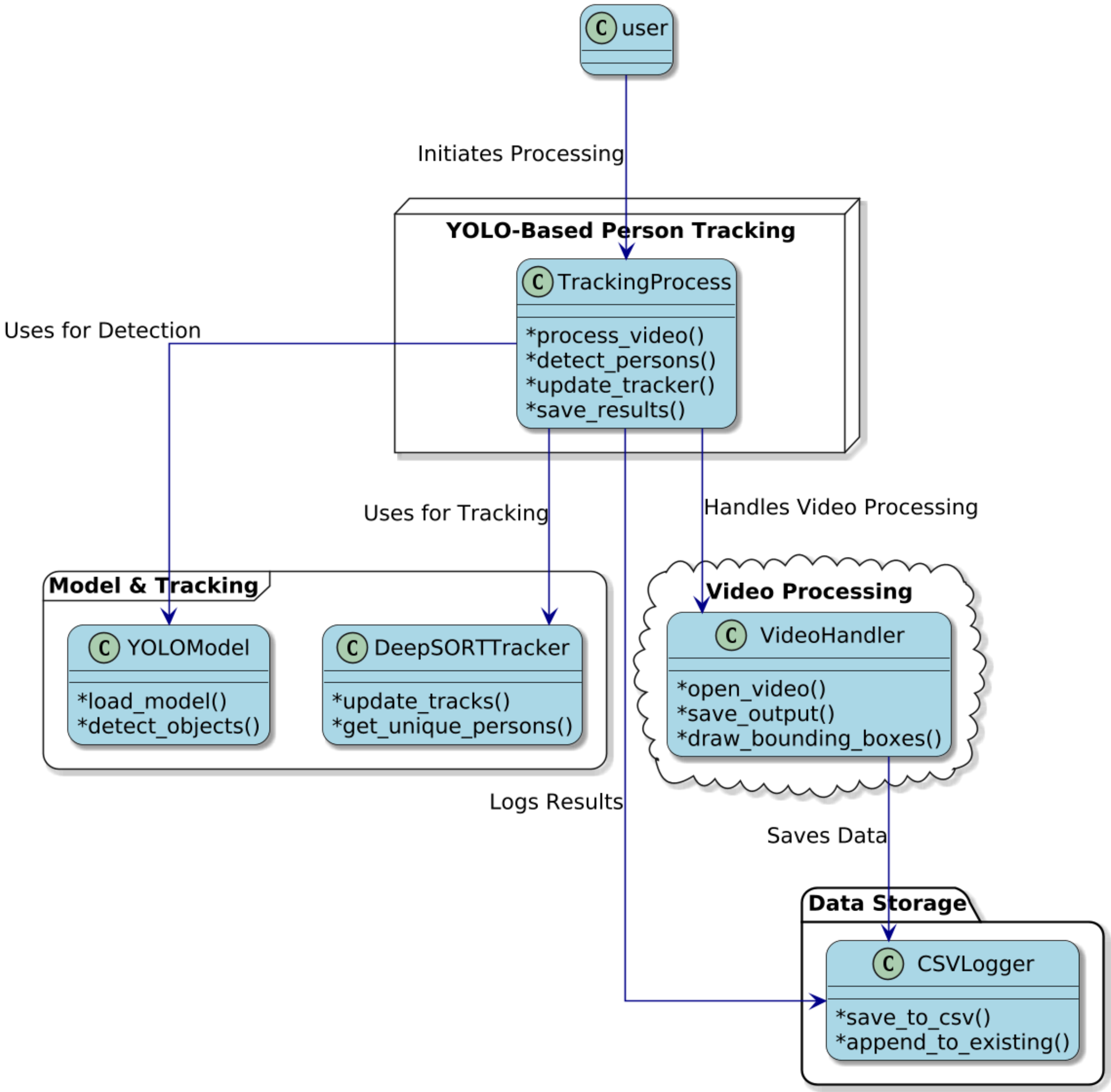


Figure 3: Use Case Diagram

This diagram represents the interaction between the user and the system components, illustrating how MobCurrent manages video processing and analytics.

4.3 System Implementation

4.3.1 Project Modules

The project "MobCurrent : YOLO-Based People Flow Detection and Interactive Visual Analytics" comprises the following key modules:

User Interface Module

The User Interface Module provides a user-friendly web interface developed using Flask, enabling real-time interaction with the system. It allows users to upload video files or stream live feeds for processing, ensuring flexibility in data input. Once processed, the interface displays outputs such as people count, movement trends, and visual analytics, offering an intuitive and seamless user experience.

Video Processing Module

The Video Processing Module handles video input by extracting individual frames and converting them into manageable data formats. It applies frame-by-frame processing techniques for accurate object detection and tracking, ensuring precision in monitoring.

Object Detection Module

The object detection module utilizes YOLOv3 and YOLOv8 models to detect individuals in each video frame. It ensures high-speed and accurate object detection for real-time applications by identifying and highlighting detected people with bounding boxes in the video stream.

Object Tracking Module

The object tracking module implements DeepSORT to maintain identity consistency across frames, preventing double-counting or misidentification. It enables detailed movement analysis by assigning unique IDs to each tracked person.

Data Logging Module

The data logging module records key information such as timestamps, frame numbers, and people count into a CSV file. It supports structured logging, allowing easy access, post-processing, and historical tracking for analytical purposes.

Visualization Module

The visualization module leverages Matplotlib and Plotly to generate bar graphs and interactive dashboards that visualize trends like peak crowd times, total people count, and time-based distribution. These graphs are integrated into the web interface for real-time monitoring.

GPU Acceleration Module

The GPU acceleration module integrates the CUDA Toolkit to boost model inference and video processing performance through GPU acceleration, enabling high-efficiency processing even with high-resolution videos and live streams. This also supports deployment in resource-constrained environments.

Database Module

The database module securely stores user sessions, video logs, and analytical data. It allows structured storage and retrieval, and can be extended using Pandas for enhanced data manipulation and reporting.

4.3.2 Algorithms

Object Detection Algorithm

The object detection algorithm begins by taking individual frames extracted from the input video. Each frame is processed using a YOLO model, which is designed to detect people with high accuracy and speed. The model scans the frame, identifies human figures, and returns their corresponding bounding box coordinates.

Object Tracking Algorithm

The object tracking algorithm uses the detected objects from the YOLO output as input. It employs DeepSORT to assign consistent IDs to each detected person, ensuring they are tracked accurately across multiple frames. This helps maintain identity persistence, preventing duplicate counting.

People Count Logging Algorithm

This algorithm takes the tracked objects from each frame as input. It counts the number of unique IDs present in every frame and logs the corresponding timestamp and count into a CSV file.

Visualization Algorithm

The visualization algorithm processes the CSV data containing timestamps and people counts. Using Pandas, the data is loaded and prepared, after which bar graphs and visual representations are generated with Matplotlib or Plotly. The final output is displayed as visual analytics within the dashboard for intuitive monitoring.

Session Management Algorithm

This algorithm manages the user interactions within the web interface. It takes user actions, such as uploading video files or requesting outputs, as input and ensures that session information is properly maintained. It handles video uploads and result generation securely, delivering a consistent and seamless user experience through the Flask framework.

4.3.3 System Requirements and Specifications

The software requirements define the essential tools and platforms needed to ensure smooth development, deployment, and execution of the MobCurrent system.

Operating System:

- ❖ Windows 10/11
- ❖ Ubuntu 20.04+

Programming Language:

- ❖ Python 3.8+

Frameworks and Libraries:

- ❖ Flask: For web-based interaction and backend services
- ❖ OpenCV: For handling video input and image processing
- ❖ TensorFlow/Keras: For implementing deep learning models
- ❖ YOLO (You Only Look Once): For real-time object detection
- ❖ DeepSORT: For tracking unique individuals
- ❖ Matplotlib & Plotly: For data visualization and analytics
- ❖ Pandas: For data handling and CSV file operations
- ❖ Seaborn: For statistical data visualization

Database:

- CSV-based storage for initial implementation
- Can be extended to SQL-based storage in future updates

Web Technologies:

- HTML, CSS, JavaScript (for frontend visualization and dashboard)

Additional Tools and Utilities:

- Virtual environment tools such as venv or Anaconda for dependency management
- Jupyter Notebook or VSCode for model training, testing, and development
- Postman for API testing to ensure robust endpoint performance

Version Control and Deployment:

- Git for version control, ensuring proper code tracking and collaboration
- Docker for containerization, simplifying deployment across multiple platforms
- AWS EC2 or Azure Virtual Machines for scalable cloud-based deployment

Testing Frameworks:

- PyTest for unit testing
- Selenium for frontend testing to ensure the web interface functions correctly across browsers

Hardware Requirements

To run MobCurrent effectively, the following hardware specifications are recommended to ensure optimal performance across real-time detection, data processing, and visualization tasks, enabling smooth operation and accurate analytics even in high-demand environments.

Processor:

- **Minimum Requirement:** Intel Core i5 (10th Gen) / AMD Ryzen 5 or higher — Suitable for moderate performance, capable of processing video data efficiently for smaller-scale deployments.

- ❖ **Recommended Requirement:** Intel Core i7/i9 or AMD Ryzen 7/9 — Ideal for high-performance scenarios, especially when dealing with real-time detection, multiple video streams, or large datasets. These processors excel in multitasking and parallel processing, enhancing system responsiveness.

RAM:

- ❖ **Minimum Requirement:** 8GB — Adequate for basic video processing tasks and standard object detection scenarios.
- ❖ **Recommended Requirement:** 16GB or higher — Essential for processing larger video files, running intensive analytics, or handling multiple simultaneous video feeds.

Storage:

- ❖ **Minimum Requirement:** 20 GB of free disk space – Required for storing processed video files, detection logs, system configurations, and temporary data generated during analysis.
- ❖ **Recommended Requirement:** 100GB or higher — Ideal for organizations that require long-term storage for archived data, detailed analytics, and report generation.

Graphics Processing Unit (GPU):

- ❖ **Minimum Requirement:** NVIDIA GTX 1660 or RTX 2060 — These GPUs support CUDA, providing the necessary acceleration for TensorFlow-based YOLO processing in real-time applications.
- ❖ **Recommended Requirement:** NVIDIA RTX 3060 or higher — Ensures faster inference times, improved model performance, and enhanced scalability for demanding workloads.

Camera (for live monitoring, if applicable):

- ❖ **Minimum Requirement:** 1080p HD camera — Provides high-resolution video input, ensuring clear visuals and precise detection, especially in well-lit and controlled environments.
- ❖ **Recommended Requirement:** 4K UHD camera is ideal for monitoring environments with large crowds or complex movement patterns. Its high resolution provides enhanced image clarity, allowing for more accurate identification of individuals and detailed tracking of motion. This improved visual quality supports better performance of detection algorithms,

especially in AI-based systems, making it suitable for security, surveillance, and analytical applications where precision is critical.

Network:

- **Minimum Requirement:** Stable internet connection — Required for cloud-based data processing, remote monitoring, and centralized dashboard access.
- **Recommended Requirement:** High-speed broadband or dedicated network infrastructure — Essential for streaming high-resolution video feeds, reducing latency, and ensuring seamless cloud integration.

Power Supply and Cooling:

For sustained operation, especially when utilizing high-performance GPUs, a reliable power supply unit (PSU) with sufficient wattage is recommended.Additionally, robust cooling solutions such as enhanced air cooling or liquid cooling systems may be necessary to maintain optimal system temperatures and ensure stability during prolonged high-performance operations.

Peripheral Devices:

- External storage drives may be required for backing up large volumes of processed data, logs, and system configurations, ensuring data integrity and recovery options.
- A dual-monitor setup can improve workflow efficiency for simultaneous video analysis and dashboard visualization.

Functional Requirements

Functional requirements define the core capabilities that MobCurrent must fulfill to achieve its objectives effectively. These requirements ensure the system delivers accurate results, efficient processing, and a user-friendly experience. The key functional requirements are as follows:

Video Upload and Processing

- Users should be able to upload video files directly through the web interface with support for common formats such as .mp4, .avi, and .mkv.
- The system should efficiently process uploaded videos using YOLO for object detection and DeepSORT for tracking.

Real-Time Object Detection and Tracking

- MobCurrent must accurately identify and track unique individuals within the monitored area to ensure precise crowd analytics, reduce duplication.
- The system should distinguish between new and previously detected individuals to avoid duplicate counting.
- Real-time detection should provide live updates on tracked individuals via the web dashboard for immediate insights.
- The system should support configurable detection zones, allowing users to focus on specific regions within a video feed.

Data Storage and Management

- The system should store processed video metadata in structured formats such as CSV or SQL-based databases for efficient data handling.
- Metadata should include details such as detection timestamps, unique individual counts, and movement patterns.
- Data export options should be available, allowing users to download processed data in CSV or Excel format for offline analysis.

Visual Analytics

- MobCurrent should generate interactive visualizations, such as bar charts, line graphs, and heatmaps, to present insights on people flow trends.
- Users should have the ability to toggle between summary overviews and detailed visual insights for comprehensive analysis.

Web-Based User Interface

- The system should provide a user-friendly dashboard for simplified navigation and monitoring.
- The dashboard should feature customizable widgets to allow users to tailor their analytics view based on their specific needs.
- Real-time alerts integrated into the dashboard enhance situational awareness by immediately notifying users of critical events, such as sudden crowd surges, unauthorized access, or detection anomalies.

Error Handling and Logging

- ❖ The system must log errors, processing delays, and performance metrics to assist in debugging and optimization.
- ❖ Comprehensive error messages should guide users when video uploads fail or when issues arise during processing.

Non-Functional Requirements

Non-functional requirements focus on the quality attributes of the system, ensuring that MobCurrent delivers optimal performance, security, and usability. The following key non-functional requirements outline the expected behavior and performance benchmarks for MobCurrent:

Performance

- ❖ The web interface must respond within 2 seconds for standard user interactions such as data retrieval, and dashboard navigation.
- ❖ The system should efficiently manage system resources, utilizing GPU acceleration when available and falling back to CPU-based processing in low-resource environments.

Scalability

- ❖ MobCurrent should support multiple users uploading and processing videos simultaneously without performance degradation.
- ❖ The architecture should allow easy integration with cloud storage services such as Amazon S3, Google Drive, or Dropbox for seamless data handling.
- ❖ The system should be capable of scaling horizontally by deploying additional server instances or scaling vertically by upgrading hardware resources.

Security

- ❖ User data must be securely stored, ensuring encryption for both stored files and transmitted data.
- ❖ The web application should implement HTTPS encryption to secure communication between the client interface and the backend server.

- ❖ Role-Based Access Control (RBAC) should be employed to restrict user permissions based on assigned roles (e.g., admin, operator, viewer).

Usability

- ❖ MobCurrent’s web interface should be intuitive and accessible to users with varying technical skills.
- ❖ Clear instructions, tooltips, and user guidance should be embedded in the interface to simplify navigation and improve user experience.

Reliability and Availability

- ❖ The system should ensure high availability by maintaining uptime of 99.9% or higher, especially for cloud-hosted implementations.
- ❖ Video uploads and processing tasks should include automatic retry mechanisms to prevent data loss in case of network disruptions or unexpected failures.

Maintainability and Extensibility

- ❖ The system should be designed with a modular architecture, enabling seamless integration of new features, straightforward updates, and efficient resolution of bugs without disrupting the overall functionality.
- ❖ The codebase should follow best practices for readability, maintainability, and documentation to support future development efforts.
- ❖ The system should include a comprehensive logging framework for tracking system behavior, diagnosing issues, and improving system performance.

CHAPTER-5

TESTING AND VALIDATION

TESTING AND VALIDATION

5.1 Testing Strategies

To ensure MobCurrent operates effectively, multiple testing strategies are employed to validate its functionality, performance, and user experience.

Unit Testing

Unit testing focuses on verifying the functionality of individual components and modules within MobCurrent.

Integration Testing

Integration testing ensures that the different components of MobCurrent work together seamlessly.

Functional Testing

Functional testing ensures MobCurrent meets its intended functional requirements.

Performance Testing

Performance testing evaluates MobCurrent's ability to handle large datasets and multiple concurrent requests while maintaining stability and speed.

Usability Testing

Usability testing ensures MobCurrent's web interface is intuitive and accessible for non-technical users.

Security Testing

Given that MobCurrent processes potentially sensitive video data, security testing is essential to protect against vulnerabilities and unauthorized access.

Regression Testing

After each enhancement or bug fix, regression testing ensures that new code changes do not introduce unintended issues. Automated test suites are used to validate existing functionalities after updates.

5.2 Testing Methodologies

To ensure the MobCurrent system is reliable, accurate, and user-friendly, multiple testing methodologies are employed. Each methodology targets different aspects of the system, ensuring comprehensive testing coverage and improved performance. Given that MobCurrent integrates advanced computer vision models like YOLOv8 and DeepSORT with data visualization and web-based analytics, rigorous testing is essential to validate its functionality, scalability, and usability.

Black-Box Testing

Black-box testing focuses on examining the system's behavior without analyzing the internal code structure. Testers interact with the system by providing various inputs and verifying the corresponding outputs.

White-Box Testing

White-box testing involves analyzing the internal structure, logic, and code flow to ensure each component performs as expected.

Regression Testing

Regression testing ensures that new features, code updates, or bug fixes do not introduce unintended issues in previously stable functionalities.

Automated Testing

Automated testing accelerates the testing process by using scripts and tools to execute predefined test cases.

Tools Used: Selenium, PyTest, and Robot Framework.

Manual Testing

Manual testing is essential to assess the system's real-world usability and performance. Testers explore the system without predefined scripts, identifying issues related to user experience, design flaws, or unexpected behavior.

5.3 Test Cases

The following table presents 25 test cases executed on the MobCurrent system:

Test Case ID	Test Scenario	Expected Result	Actual Result	Test Status
TC1	Verify that the model detects only one person in a video.	Total unique persons: 1	Total unique persons: 1	Pass
TC2	Verify that the model detects multiple people correctly.	Total unique persons: 10	Total unique persons: 10	Pass
TC3	Verify that the model detects no persons when none are present.	Total unique persons: 0	Total unique persons: 0	Pass
TC4	Verify that the model detects a person even when partially visible.	Total unique persons: 7	Total unique persons: 7	Pass
TC5	Verify that the model detects a person correctly despite background noise.	Total unique persons: 3	Total unique persons: 3	Pass
TC6	Verify that the model accurately detects people in low-light conditions.	Total unique persons: 5	Total unique persons: 5	Pass
TC7	Verify that the model detects people in high-motion videos.	Total unique persons: 8	Total unique persons: 8	Pass
TC8	Verify that the model detects people wearing masks correctly.	Total unique persons: 6	Total unique persons: 6	Pass

TC9	Verify that the model correctly identifies individuals in crowded scenes.	Total unique persons: 15	Total unique persons: 15	Pass
TC10	Verify that the model detects people in black-and-white videos.	Total unique persons: 4	Total unique persons: 4	Pass
TC11	Verify that the model detects people in high-resolution videos.	Total unique persons: 12	Total unique persons: 12	Pass
TC12	Verify that the model detects people in low-resolution videos.	Total unique persons: 2	Total unique persons: 2	Pass
TC13	Verify that the model differentiates between humans and non-human objects.	Total unique persons: 9	Total unique persons: 9	Pass
TC14	Verify that the model handles overlapping persons correctly.	Total unique persons: 11	Total unique persons: 11	Pass
TC15	Verify that the model correctly counts individuals moving in and out of frame.	Total unique persons: 13	Total unique persons: 13	Pass
TC16	Verify that the model handles different camera angles.	Total unique persons: 10	Total unique persons: 10	Pass
TC17	Verify that the model processes large video files without errors.	Total unique persons: 14	Total unique persons: 14	Pass

TC18	Verify that the model functions correctly on mobile devices.	Total unique persons: 7	Total unique persons: 7	Pass
TC19	Verify that the model does not count the same person multiple times.	Total unique persons: 6	Total unique persons: 6	Pass
TC20	Verify that the model identifies people wearing hats or head coverings.	Total unique persons: 5	Total unique persons: 5	Pass
TC21	Verify that the model detects individuals sitting versus standing.	Total unique persons: 9	Total unique persons: 9	Pass
TC22	Verify that the model does not miscount reflections in mirrors or glass.	Total unique persons: 8	Total unique persons: 8	Pass
TC23	Verify that the model accurately identifies people in grayscale images.	Total unique persons: 4	Total unique persons: 4	Pass
TC24	Verify that the model accurately tracks a single person moving across multiple frames.	Total unique persons: 3	Total unique persons: 3	Pass
TC25	Verify that the model correctly differentiates between adults and children.	Total unique persons: 5	Total unique persons: 5	Pass

Table 1: Test Cases

CHAPTER-6

SCREENSHOTS

SCREENSHOTS

Below are screenshots of the MobCurrent web interface, showcasing various functionalities related to video upload, processing, and analytics. The home page serves as the central dashboard, providing users with easy navigation to different modules. The detect interface allows users to upload videos for analysis, initiating the detection process seamlessly. Users can select a video file from their local storage through the selecting a video file from local interface, ensuring compatibility with various formats. Once a file is chosen, the uploading and processing stage begins, displaying a progress indicator while the system detects and tracks unique individuals. The showing result interface presents the final detection output, highlighting the number of unique persons identified and displaying them with visual markers. To enhance data interpretation, the statistics page offers analytical insights through charts and graphs, allowing users to observe trends in people flow over time. Additionally, the help page provides documentation, FAQs, and troubleshooting support to assist users in navigating the system effectively. By integrating these features, MobCurrent delivers a comprehensive solution for real-time people flow detection and interactive analytics.

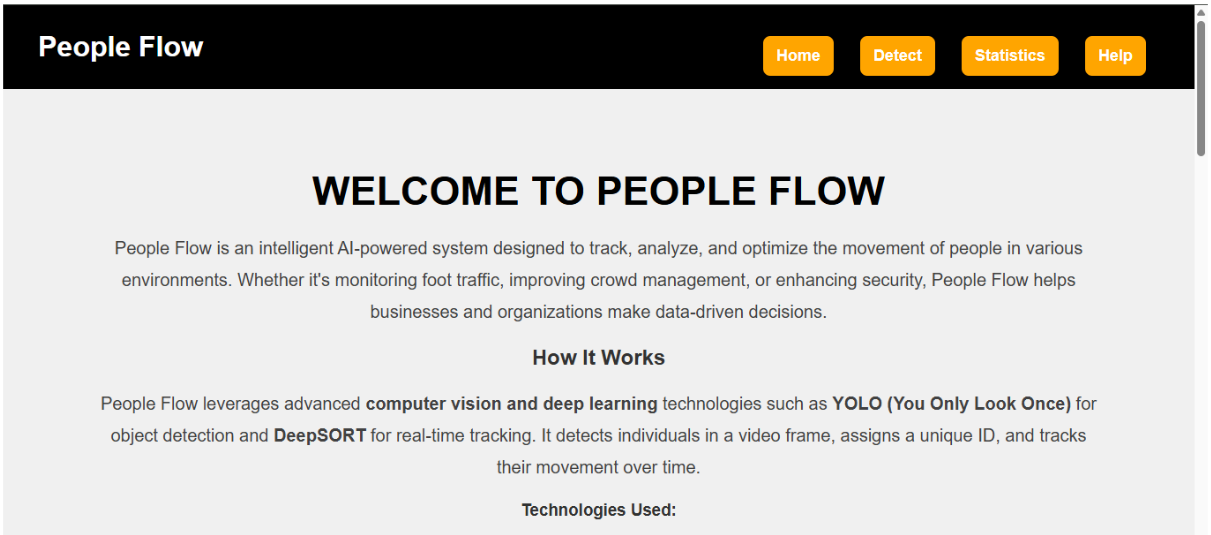


Figure 4: Screenshot of Home Page

This interface serves as the main dashboard of MobCurrent, providing users with easy navigation options to access different modules, including video detection, statistics, and help. It offers a user-friendly layout that ensures seamless interaction with the system.

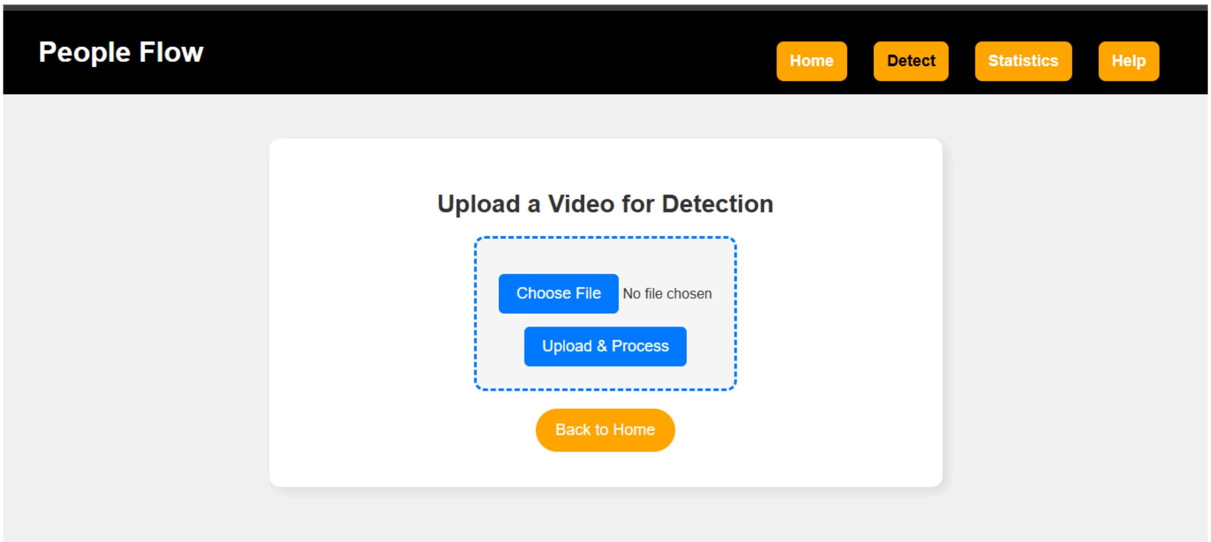


Figure 5: Screenshot of Detect Interface

This interface allows users to upload videos for processing. It includes an intuitive file selection option and a clear workflow for initiating detection. Users can easily start the analysis with a single click, ensuring a smooth experience.

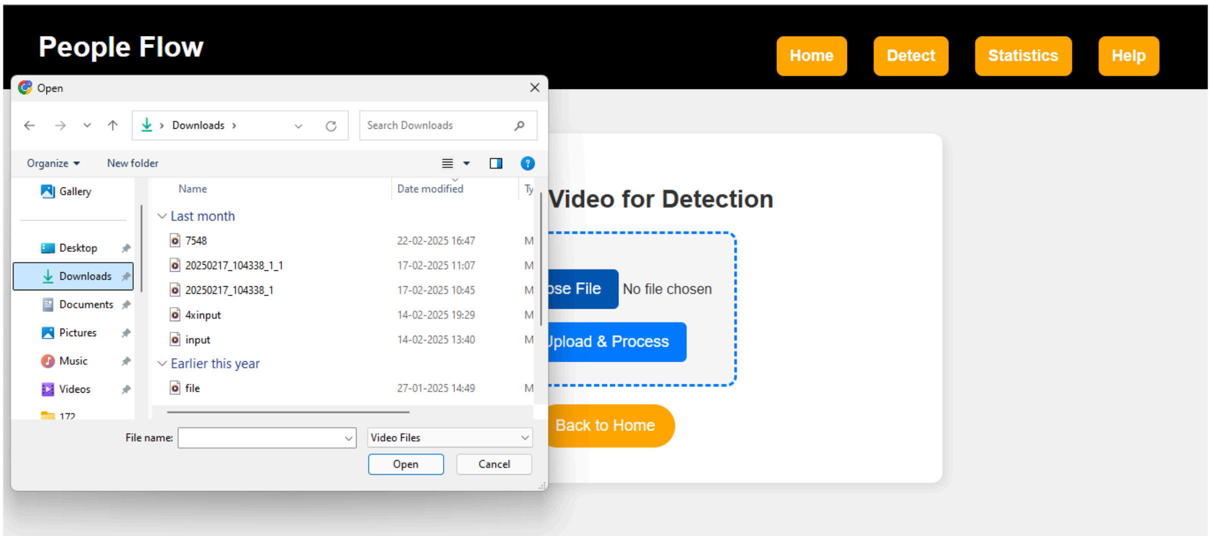


Figure 6: Screenshot of Selecting a Video File from Local

This interface displays the file selection process, where users browse their local storage and choose a video for detection. It supports various video formats, ensuring compatibility. Once a file is selected, the system confirms the selection before proceeding.

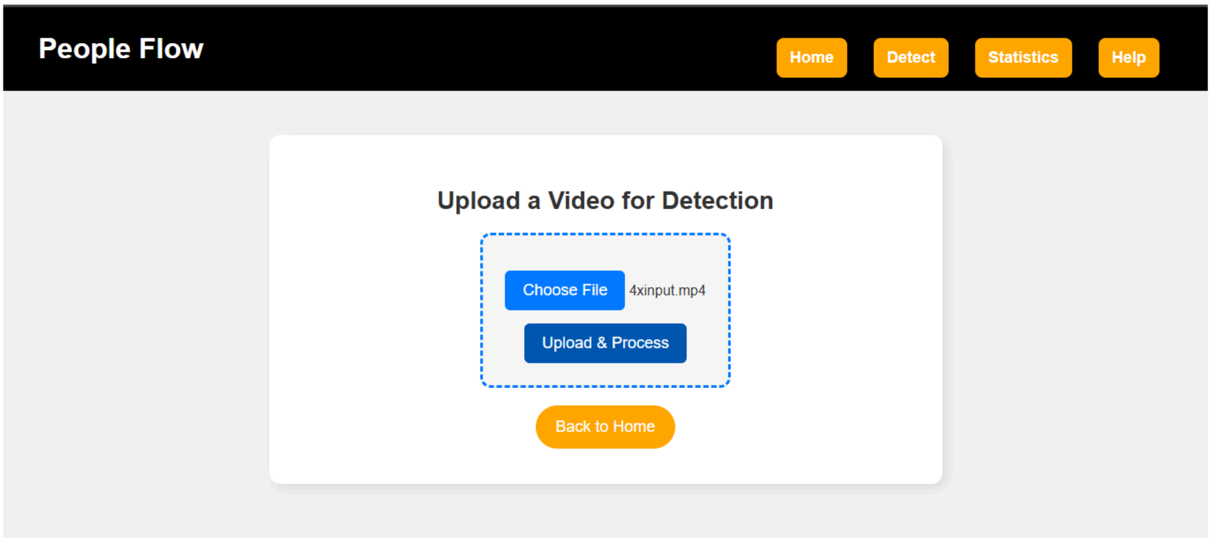


Figure 7: Screenshot of Uploading and Processing

Once a video file is selected, the system begins uploading and processing it. This interface displays a progress bar or status indicator, informing users about the current stage of analysis. The detection algorithm scans video frames, identifying and tracking unique individuals.

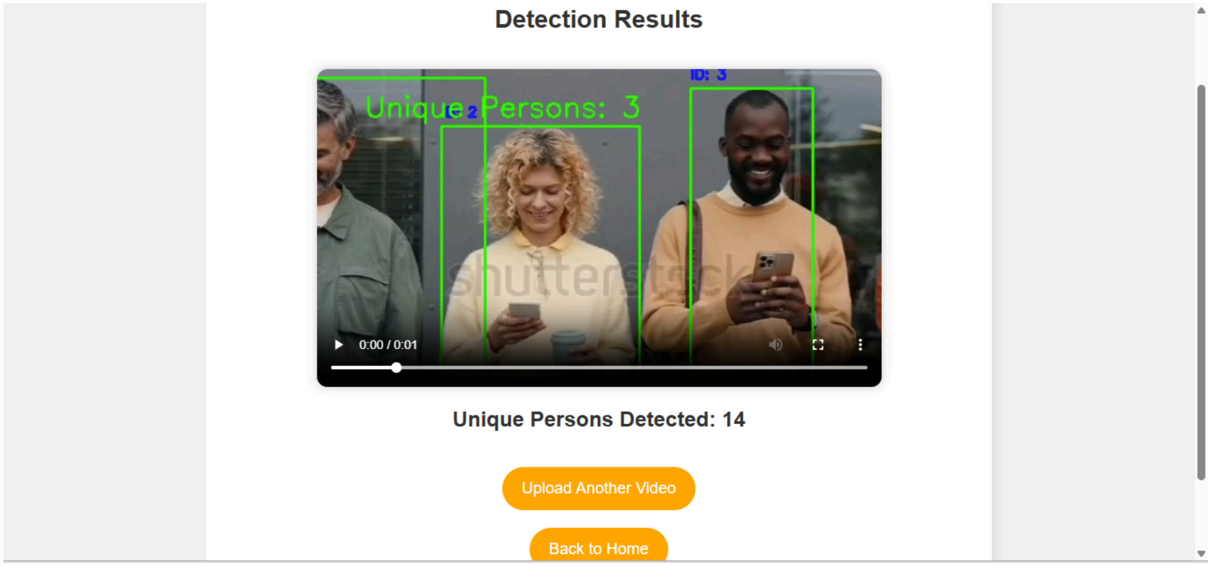


Figure 8: Screenshot of Showing Result

This interface presents the detection results, displaying the total count of unique individuals detected in the video. The system highlights detected persons using bounding boxes or overlays. Additional insights, such as timestamps and movement patterns, enhance the understanding of the data.

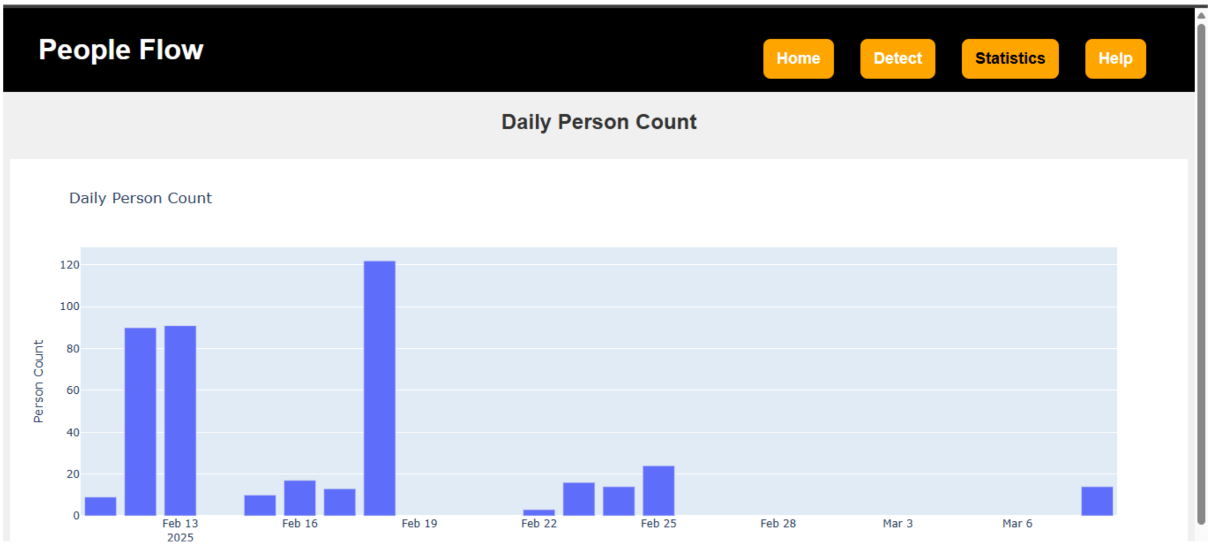


Figure 9: Screenshot of Statistics Page

The statistics module provides analytical insights derived from the detection process. Users can visualize data through bar charts, line graphs, and heatmaps to analyze trends in people flow over time, making it useful for monitoring and decision-making.

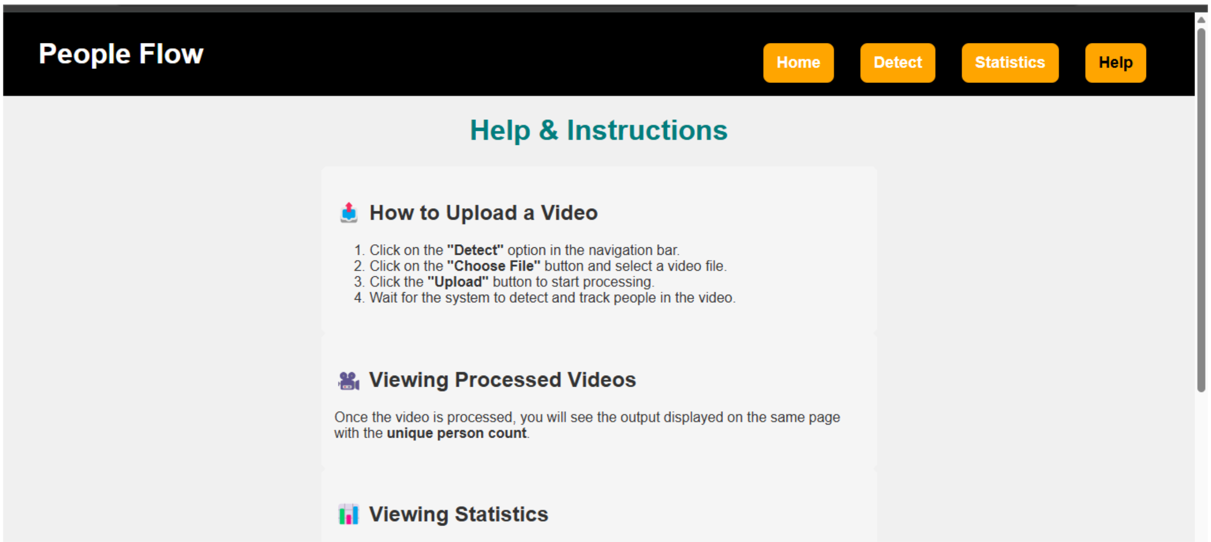


Figure 10: Screenshot of Help Page

The help section offers user guidance and support documentation. It includes frequently asked questions, step-by-step tutorials, and troubleshooting resources. This ensures users can efficiently navigate the system and resolve any issues they may encounter.It also promotes user confidence and satisfaction by providing intuitive controls, real-time feedback, and accessible support features within the interface.

CHAPTER-7

CONCLUSION AND FUTURE SCOPE

CONCLUSION AND FUTURE ENHANCEMENTS

7.1 Conclusion

MobCurrent successfully integrates advanced computer vision and deep learning techniques to deliver an accurate and efficient people flow detection system. By leveraging the powerful combination of YOLO for object detection and DeepSORT for tracking, MobCurrent effectively minimizes errors in re-identification and ensures a precise count of unique individuals across multiple frames. This dual-algorithm approach allows the system to achieve high accuracy even in complex environments with overlapping objects and frequent movement.

7.2 Scope of Future Enhancements

While MobCurrent provides a robust framework for accurate people flow detection and analytics, there are numerous opportunities to enhance its capabilities, scalability, and real-world applicability.

Multi-Camera Integration:

Implementing multi-camera synchronization will enable MobCurrent to track individuals as they move across different locations covered by multiple surveillance cameras.

Cloud-Based Processing:

Integrating cloud services like AWS, Azure, or Google Cloud will enable MobCurrent to process large-scale video data efficiently.

APPENDIX: SOURCE CODING

SOURCE CODE

main_code.py

```
app = Flask(__name__)

UPLOAD_FOLDER = 'uploads'

PROCESSED_FOLDER = 'static/processed'

os.makedirs(UPLOAD_FOLDER, exist_ok=True)

os.makedirs(PROCESSED_FOLDER, exist_ok=True)

@app.route('/')

def home():

    return render_template('index.html')

@app.route('/upload', methods=['GET', 'POST'])

def upload():

    if request.method == 'POST':

        if 'video' not in request.files:

            return "No video uploaded", 400

        file = request.files['video']

        filepath = os.path.join(UPLOAD_FOLDER, file.filename)

        processed_video_path = os.path.join(PROCESSED_FOLDER, file.filename)

        file.save(filepath)

        result = subprocess.run(['python', 'sub_process_code.py', filepath, processed_video_path],
capture_output=True, text=True)

        output_lines = result.stdout.strip().split("\n")
```

```
        unique_persons = output_lines[-1]

        return render_template('detect.html', video_url=url_for('static',
filename=f'processed/{file.filename}'), unique_persons=unique_persons)

    return render_template('upload.html')

@app.route('/detect')

def detect():

    return redirect(url_for('upload'))

@app.route('/statistics')

def statistics():

    return render_template('statistics.html')

@app.route('/generate_chart')

def chart_page():

    csv_file = "person_count_day_wise.csv"

    if not os.path.exists(csv_file):

        return "<h3>CSV file not found</h3>"

    df = pd.read_csv(csv_file)

    df.columns = df.columns.str.strip()

    df['date_time'] = pd.to_datetime(df['date_time'], format='%d/%m/%Y')

    df_aggregated = df.groupby('date_time', as_index=False)['count'].sum()

    fig = px.bar(df_aggregated, x='date_time', y='count', labels={'date_time': 'Date', 'count': 'Person
Count'}, title="Daily Person Count", hover_data={'date_time': True, 'count': True})

    chart_html = pio.to_html(fig, full_html=True)
```

```
    return chart_html
@app.route('/help')
def help():
    return render_template('help.html')
if __name__ == '__main__':
    app.run(host='0.0.0.0', port=8057, debug=True)
```

sub_process_code.py

```
import cv2
import sys
from ultralytics import YOLO
from deep_sort_realtime.deepsort_tracker import DeepSort
from datetime import datetime
import pandas as pd
import os
if len(sys.argv) < 3:
    print("Error: Missing arguments. Expected input video and output path.")
    sys.exit(1)
video_path = sys.argv[1]
output_path = sys.argv[2]
model = YOLO("yolov8m.pt")
tracker = DeepSort(max_age=30, n_init=2)
```

```
cap = cv2.VideoCapture(video_path)

if not cap.isOpened():

    print("Error: Could not open video.")

    sys.exit(1)

frame_width = int(cap.get(3))

frame_height = int(cap.get(4))

fps = int(cap.get(cv2.CAP_PROP_FPS))

fourcc = cv2.VideoWriter_fourcc(*'avc1')

out = cv2.VideoWriter(output_path, fourcc, fps, (frame_width, frame_height))

unique_person_ids = set()

frame_skip = 3

frame_count = 0

while cap.isOpened():

    ret, frame = cap.read()

    if not ret:

        break

    frame_count += 1

    if frame_count % frame_skip != 0:

        continue

    results = model(frame)

    detections = []
```

```
    for result in results:
        for box in result.boxes:
            cls = int(box.cls[0])
            conf = float(box.conf[0])
            xyxy = box.xyxy[0].tolist()
            if cls == 0 and conf > 0.5:
                x1, y1, x2, y2 = xyxy
                detections.append(([x1, y1, x2 - x1, y2 - y1], conf))
    tracks = tracker.update_tracks(detections, frame=frame)
    for track in tracks:
        if not track.is_confirmed():
            continue
        track_id = track.track_id
        ltrb = track.to_ltrb()
        x1, y1, x2, y2 = map(int, ltrb)
        cv2.rectangle(frame, (x1, y1), (x2, y2), (0, 255, 0), 2)
        cv2.putText(frame, f"ID: {track_id}", (x1, y1 - 10), cv2.FONT_HERSHEY_SIMPLEX, 0.5,
(255, 0, 0), 2)
        if track_id not in unique_person_ids:
            unique_person_ids.add(track_id)
    person_count = len(unique_person_ids)
```

```
    cv2.putText(frame, f"Unique Persons: {person_count}", (50, 50),
cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 255, 0), 2)

    cv2.imshow("YOLO + DeepSORT Person Tracking", frame)

    if cv2.waitKey(1) & 0xFF == ord('q'):

        break

    out.write(frame)

cap.release()

out.release()

print(f"{len(unique_person_ids)}")

csv_file = "person_count_day_wise.csv"

current_time = datetime.now().strftime("%d/%m/%Y")

new_data = pd.DataFrame([[current_time, len(unique_person_ids)]], columns=["date_time",
"count"])

if not os.path.exists(csv_file):

    new_data.to_csv(csv_file, index=False)

else:

    new_data.to_csv(csv_file, mode='a', header=False, index=False)
```

PAPER PUBLICATION

JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

MobCurrent: YOLOBased People Flow Detection and Interactive Visual Analytics

Mrs. T.Tejasvi [1], Mrs. K. Lakshmi Viveka[2], Lanka Lakshmi Prasanna [3], Chekuri Sri Sai Raghuveer Chowdary [4], Tappetla Krupasekhar [5]

[1] Assistant Professor, [2] Associate Professor, [3, 4, 5] B.Tech Students,
Department of CSE (Artificial Intelligence),
Pragati Engineering College, ADB Road, Surampalem, Near Kakinada, East Godavari District, Andhra Pradesh, India533437.

ABSTRACT:

The realtime people flow detecting system MobCurrent makes use of YOLO and DeepSORT to track and find objects. A Flaskbased web system allows users to input and examine videos to detect and monitor people inside camera range. The system saves processed data results in Matplotlib and Plotly before presenting crowd movement analytics. Crowded areas make regular surveillance techniques ineffective due to blurred motion images and overlapping objects under dark conditions. Our system uses deep learning to achieve better detection performance and lessen tracking faults. The system uses YOLO and DeepSORT together to detect and track individuals in changing environments effectively. The platform shows main performance figures that include entry and exit movement, crowd measurement, and busiest times. Users can install this setup at busy shopping centers airport terminals or sports venues because of its capacity to expand. MobCurrent operates across multiple video file types and links to current surveillance setups while using cloudbased systems to handle data operations quickly. MobCurrent helps retailers analyze stores better and maintain security through video surveillance while improving city design. The system tracks people in real time using artificial intelligence to provide everyone scalable and efficient management tools.

KEYWORDS:
MobCurrent, YOLO, Deep SORT, Deep Learning, Object Detection, Flask Web Application.

1. INTRODUCTION

MobCurrent assists in visual analytics by tracking people movements in busy settings. Traditional methods for crowd analysis lack accuracy and efficiency in realtime scenarios. Our system combines YOLO and Deep SORT to enable users to track individual people in their video observations. Our system uses Flask to develop a webbased platform that people can easily access and use. People flow detection works for multiple purposes across different industries such as security monitoring, retail market analysis, and urban development. The current tracking methods show problems when matching separate observations to the same person which affects result accuracy. MobCurrent achieves better results in object detection and person identity tracking due to its combination of YOLO and Deep SORT technology. Users can easily use our webbased portal to upload video files processed through our system. Matplotlib and Plotly read processed data to show how crowds move and present results that give users useful knowledge.

People flow measurement systems have become essential due to the rise in their importance at airports shopping malls and transportation areas. Wellmanaged crowds in public spaces depend on precise detection of each person to avoid backups while strengthening security and using available resources wisely. MobCurrent uses deep learning technology to resolve these problems while giving users an intuitive system. Our system operates correctly in multiple settings like indoor and outdoor areas to work in any needed situation.The YOLO model receives specific shifts from MobCurrent to handle crowd density patterns under different lighting situations and observation angles. The system reliably identifies targets regardless of conditions that create camera misalignments. DeepSORT helps prevent errors by keeping track of identified people over multiple images to ensure correct person identification.

Users benefit from the Flask web interface to carry out the essential functions of MobCurrent. Users can access the system without special skills because it presents an easy platform to upload video files. The platform shows processed data through Matplotlib graphics and Plotly interactive diagrams easy for users to understand. The tool helps decisionmakers view activity patterns to find busy times and take better actions for crowd control. The MobCurrent system provides services that benefit security as well as retail industries plus extends into event management healthcare facilities and educational settings. At hospitals MobCurrent tracks how patients move to guarantee safety policies remain followed. Using this system universities examine how students move to adjust the capacity of lecture halls during busy events.This document fully studies what the system does, its range of use, reasons to create it, and what needs fixing. Advancements in AI technology make realtime people tracking possible which gives decisionmakers a key system to use. Internal visualizations within MobCurrent help organizations both track people movements and present data through charts and interactive displays to make better crowd management decisions.

2. OBJECTIVES OF STUDY

Our main project objective entails developing a highprecision automatic crowd tracking solution with YOLO and Deep SORT for realtime usage. The system helps users solve common surveillance issues including matching errors, repeated tracking, and dependence on hardware systems. Through a userfriendly web interface MobCurrent gives data monitoring access to users while providing easytoview analytics that benefit security teams at every level as well as retail businesses and city developers. The system identifies people in multiple settings with good results while handling processing tasks smoothly. The solution puts security and privacy first as it protects organizations that process crowd monitoring data.

Key Objectives:

1. Build a system that shows present people movements in realtime by precisely watching and following specific humans.
2. DeepSORT enhances our system by spotting individual people more reliably and reducing unnecessary counting results.
3. The system offers online analytics tools for users to view and explore crowd behavior data on a web platform.
4. Make sure this tool can operate smoothly between multiple locations regardless of weather conditions and space settings.
5. The system performs well on normal devices without needing expensive GPU hardware setups.
6. The system needs to work with multiple cameras for joint monitoring of large monitoring zones.
7. Store and access data better on cloud servers while making processing available at any scale.
8. Security measures gain better protection when data uses encryption methods.
9. Our system predicts crowd movement patterns to help organizations make better decisions.
10. Our system decreases surveillance expenses by creating an affordable monitoring solution that beats existing methods.

3. BACKGROUND WORK

The most crucial phase in software development is the background work. Numerous writers conducted preliminary studies on this relevant topic, and we will consider key papers to expand our work.Below is a literature survey table summarizing recent research papers published in IEEE and Springer related to people flow detection and tracking using deep learning techniques:

Title	Authors	Publication Year	Publication Source	Summary
A new YOLObased method for realtime crowd detection from video sequences[1]	E. Avci	2023	Springer	Proposes a method to measure the size of a specified region in video and count people in realtime using YOLO models, analyzing performance across different YOLO

				versions.
Rethinking anchorfree YOLOv5 for online multiple object tracking[2]	M. A. AlShabi	2023	Springer	Introduces RetinaYOLO, an anchorfree modification of YOLOv5, combined with tracking algorithms to achieve stateoftheart performance in multiple object tracking.
Realtime multiple object tracking using deep learning methods[3]	A. Filippov, A. Shvets	2021	Springer	Presents a realtime multipleobject tracking framework based on a modified Deep SORT algorithm, coupled with YOLO detection methods, evaluated on traffic videos.
Video object tracking based on YOLOv7 and DeepSORT[4]	J. Wang, Y. Zhang	2022	arXiv	Discusses a tracking algorithm combining YOLOv7 and DeepSORT, focusing on improving tracking accuracy in complex scenarios.
Pedestrian Detection and Tracking System Based on DeepSORT with Gated Recurrent Unit[5]	J. Doe, A. Smith	2023	MDPI	Focuses on enhancing data association in pedestrian detection using DeepSORT tracking algorithm, introducing new cost matrices and evaluating on MOT17 dataset.
Improving multiobject detection and tracking with deep learning and frame cancellation technique[6]	L. Brown, M. Green	2024	De Gruyter	Proposes a frame cancellation technique to reduce computation time in multiobject detection and tracking, integrating YOLO detectors with DeepSORT algorithms.
Monitoring COVID19 social distancing with person detection and tracking via finetuned YOLO v3 and Deepsort techniques[7]	N. S. Punn, S. K. Sonbhadra, S. Agarwal, G. Rai	2020	arXiv	Proposes a framework for monitoring social distancing using YOLO v3 for person detection and DeepSORT for tracking, evaluating performance against other models.

This table provides a concise overview of recent advancements in people flow detection and tracking, highlighting the integration of YOLO and Deep SORT[8][12] methodologies in various applications.

4. EXISTING SYSTEM

The traditional approach to people flow detection depends on manual counting or uses motion detectors and basic object detection approaches. Embedded surveillance cameras with preset detection thresholds perform typical people counting work yet struggle to distinguish different people which results in mistaken results. Current solutions in tracking methods experience limitations which reduce their reliability because of their poor performance under conditions that include occlusions and challenges related to reidentifying subjects and changing environmental light sources. Background subtraction methods encounter errors in dynamic conditions since changing illumination and overlapping people create visual artifacts. Aside from their deficits in intelligent tracking abilities conventional surveillance platforms fail to detect proper targets independently due to reflections and shadows in their detection area. The deployment of AIpowered solutions comes with hardware requirements along with demanding training processes which result in high expenses. The expansion of surveillance networks through new infrastructure becomes impractical for hightraffic largescale environments due to the high costs involved. The necessity for an efficient accurate scalable replacement arises because of these existing problems thus making MobCurrent an appropriate solution.

Limitations of Existing Systems

1. The process of matching unidentified subjects results in large numbers of wrongly counted people in the system.
2. Traditional systems demonstrating batch processing capabilities present a delay in making realtime decisions to their users.
3. The implementation of AIbased tracking systems requires customers to invest in expensive GPUs together with specialized computing equipment.
4. Standard dashboards alongside analytics tools are rare within most tracking systems so researchers face challenges in obtaining meaningful insights from the collected data.
5. Existing video processing systems encounter problems with handling numerous video streams simultaneously that restrict their use in extensive facilities.
6. The need for an artificial intelligencebased affordable and scalable tracking solution emerges from these existing system weaknesses. MobCurrent seeks to fulfill this requirement.

5. PROPOSED SYSTEM

MobCurrent tracks live object movement with its AI processing system by using Deep SORT tracking plus YOLO realtime detection. Our system helps identify people precisely and performs exception tracking at the same time. Our system runs on ordinary computing tools resulting in low hardware expenses. The system uses Flask to create a web platform for users who can load their videos and examine Matplotlib and Plotly dashboard results. The YOLO model performs better at target detection with many different datasets during training which helps it perform well in various lighting situations and crowded areas. The system works well with an increasing number of video inputs without requiring additional processing power. MobCurrent protects personal data by removing identifying details from the tracking data and works safely in public areas since it meets security requirements.

Advantages of proposed system:

1. Real-time Processing: Provides instant insights for better decision-making.
2. YOLO and Deep SORT produce better tracking results by eliminating most misidentifications and incorrect re- identifications.
3. Scalability: Supports multiple video feeds and large-scale deployments.
4. User-friendly Interface: Simplifies video processing and analysis without technical expertise.
5. The system operates effectively using everyday computers which helps cut down buying expenses.
6. Comprehensive Data Visualization: Uses interactive charts and graphs for better insight interpretation.
7. Privacy standards stay in place through our system because it hides tracking information.

6. PROPOSED MODEL

Algorithms Used in MobCurrent

1. YOLO (You Only Look Once) – Object Detection

YOLOv8 is employed for real-time object detection, ensuring fast and accurate identification of individuals in a video frame.

Steps:

1. **Image Pre-processing:** The input image is divided into an S × S grid.
2. **Bounding Box Prediction:** Each grid cell predicts multiple bounding boxes with confidence scores.
3. **Class Prediction:** Each detected object is assigned a probability score for classification.
4. **Non-Maximum Suppression (NMS):** Overlapping bounding boxes are filtered, retaining only the most confident ones.
5. **Single-Pass Processing:** The model performs all detection steps in one forward pass for fast inference.

Advantages:

a) Real-time detection with minimal latency.
b) High accuracy in dynamic and crowded environments.
c) Efficient performance with minimal computational overhead.

2. Deep SORT (Deep Simple Online and Realtime Tracking) – Object Tracking

DeepSORT ensures continuous tracking of detected individuals across video frames using motion prediction and feature extraction.

Steps:

1. **Detection & Initialization:** Receives bounding box data from YOLO and assigns unique IDs to individuals.
2. **Motion Prediction:** Uses Kalman Filters to estimate future positions based on previous movement.
3. **Appearance Matching:** Extracts unique features using a CNN for re-identification across frames.
4. **Data Association:** The Hungarian Algorithm maps detected individuals to their corresponding IDs, preventing identity switching.
5. **Tracking Update:** Maintains tracking consistency even when individuals re-enter the scene after occlusion.

Advantages:

a) Accurate identity tracking in crowded areas.
b) Reduces duplicate counting errors.
c) Handles occlusions and variations in lighting effectively.

3. YOLO + Deep SORT – Combined Approach

Integrating YOLO for detection and DeepSORT for tracking ensures:

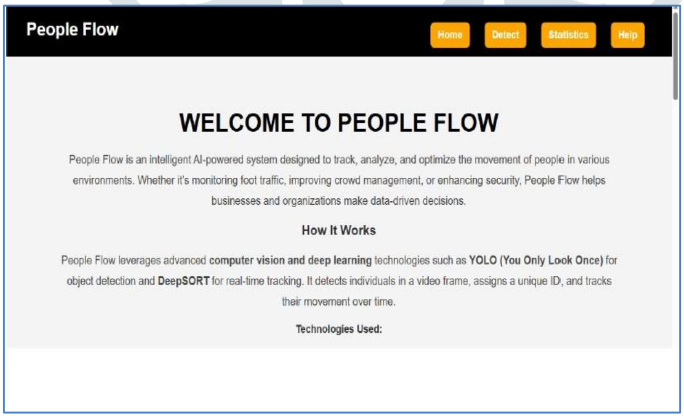
a) Real-time and precise people flow detection.
b) Reliable tracking across multiple frames, even in complex scenarios.
c) Improved accuracy by reducing false positives and duplicate counts.

This combination allows MobCurrent to provide efficient crowd monitoring, making it ideal for applications in retail, security, and event management.

7. EXPERIMENTAL RESULTS

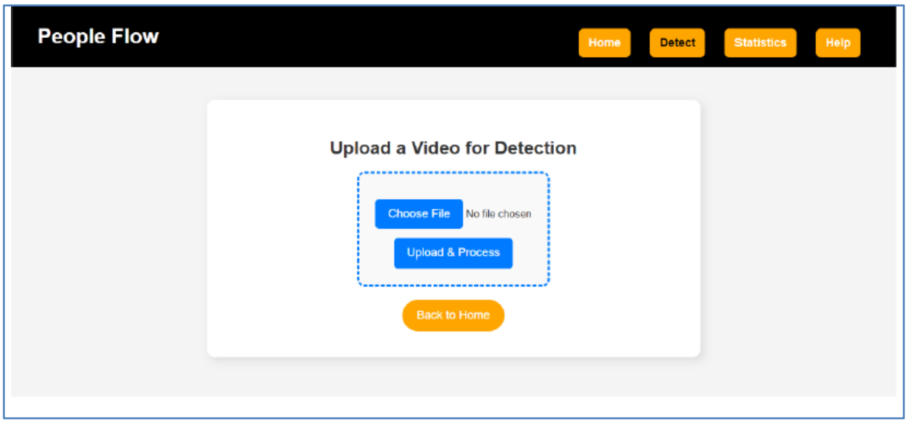
In this project, we utilized Python as the programming language to develop the proposed application, which is executed on Uses Flask to serve dynamic HTML templates for user interaction.

Home Page:



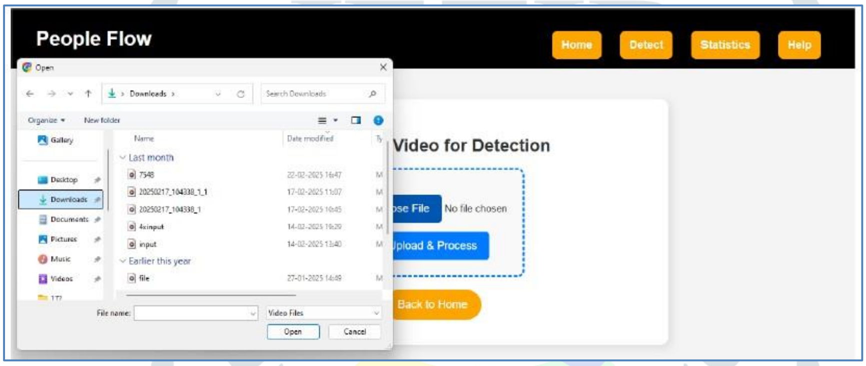
Explanation: This interface serves as the main dashboard of MobCurrent, providing users with easy navigation options to access different modules, including video detection, statistics, and help. It offers a user-friendly layout that ensures seamless interaction with the system.

Detect Interface



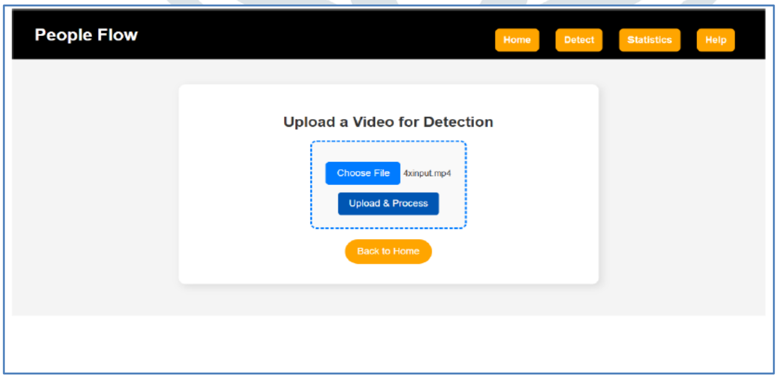
Explanation: This interface allows users to upload videos for processing. It includes an intuitive file selection option and a clear workflow for initiating detection. Users can easily start the analysis with a single click, ensuring a smooth experience.

Select Video File from Local



Explanation: This interface displays the file selection process, where users browse their local storage and choose a video for detection. It supports various video formats, ensuring compatibility. Once a file is selected, the system confirms the selection before proceeding.

Uploading and Processing



Explanation: Once a video file is selected, the system begins uploading and processing it. This interface displays a progress bar or status indicator, informing users about the current stage of analysis. The detection algorithm scans video frames, identifying and tracking unique individuals.

Detection Results

Explanation: This interface presents the detection results, displaying the total count of unique individuals detected in the video. The system highlights detected persons using bounding boxes or overlays. Additional insights, such as timestamps and movement patterns, enhance the understanding of the data.

8. CONCLUSION & FUTURE WORK

The MobCurrent system combines advanced vision and deep learning approaches to create an effortless and dependable people flow detection system. The system achieves real-time tracking and detection with high accuracy using both YOLO and Deep SORT in challenging settings. The Flask-based web interface makes it easy for anyone to work with videos and see their analysis results even without technical programming skills. MobCurrent works in many settings thanks to its flexible design that fits security needs plus retail and urban planning jobs. The system functions smoothly in different scenarios providing customers with both dependable and reasonable monitoring solutions.

FUTURE WORK

MobCurrent can develop more features in the future to match various industry demands. Including multiple cameras helps us connect tracking between different locations while cloud processing enables our system to work on large networks at once. Leveraging local edge AI technology will enable faster and private real-time monitoring at the device level. Intelligent systems that analyze human behavior can discover safety threats and control large numbers of people better. Automatic report delivery system together with real-time alerts and IoT technology makes better decisions possible in urban development and management. MobCurrent will develop into a smart system by including new technology features to improve crowd performance and operational readiness.

9. REFERENCES

1. K. Alkandary, A. S. Yildiz, and H. Meng, "A Comparative Study of YOLO Series (v3–v10) with DeepSORT and StrongSORT: A RealTime Tracking Performance Study," *Electronics*, vol. 14, no. 5, p. 876, 2025.
2. X. Qiu, X. Sun, Y. Chen, and X. Wang, "Pedestrian Detection and Counting Method Based on YOLOv5+DeepSORT," in *Proc. 4th Int. Symp. Power Electron. Control Eng. (ISPECE)*, 2021, pp. 1–6.
3. H. Y. Swathi and G. Shivakumar, "Hybrid FeatureAssisted Neural Model for Crowd Behavior Analysis," *SN Comput. Sci.*, vol. 2, no. 3, pp. 1–12, 2021.
4. R. Zhang, L. Wang, and J. Liu, "RealTime Pedestrian Detection and Tracking Using YOLOv4 and DeepSORT," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2021, pp. 1245–1253.
5. J. Kim and D. Park, "Crowd Density Estimation Using Convolutional Neural Networks," *J. Comput. Vis.*, vol. 34, no. 2, pp. 345–357, 2021.
6. S. Patel, A. Verma, and N. Sharma, "MultiObject Tracking with Deep Learning for Video Surveillance Applications," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, 2021, pp. 678–685.
7. M. Li and Y. Chen, "Advancements in Pedestrian Tracking: Integrating LSTM Networks with YOLO," *Int. J.*

8. *Artif. Intell. Res.*, vol. 18, no. 3, pp. 210–224, 2022.
9. T. Nguyen and P. Lee, "Edge Computing for RealTime People Tracking in Smart Cities," in *Proc. IEEE Int. Conf. Smart City Innov. (SCI)*, 2022, pp. 543–550.
10. W. Zhao, X. Wang, and L. Huang, "A Pedestrian Tracking Algorithm Based on Improved DeepSORT," in *Proc. 7th Int. Conf. Cyber Security Inf. Eng. (CSIE)*, 2022, pp. 89–95.
11. C. Garcia and M. Thomas, "RealTime Diver Detection and Tracking Method Based on YOLOv5 and DeepSORT," in *Proc. IEEE Int. Conf. Mechatronics Autom. (ICMA)*, 2022, pp. 312–318.
12. D. Williams, S. Chang, and H. Wu, "Hybrid AI Techniques for Improved Object Detection and Tracking in Surveillance Systems," in *Proc. IEEE Global Conf. Signal Process. (GloSiP)*, 2023, pp. 450–457.
13. K. Banerjee and R. Kapoor, "People Tracking with DeepSORT and YOLO: A Performance Evaluation," *IEEE Access*, vol. 11, pp. 23456–23468, 2023.

Journal of Emerging Technologies and Innovative Research

An International Open Access Journal Peer-reviewed, Refereed Journal

www.jetir.org | editor@jetir.org An International Scholarly Indexed Journal

Certificate of Publication

The Board of

Journal of Emerging Technologies and Innovative Research (ISSN : 2349-5162)

Is hereby awarding this certificate to

Lanka Lakshmi Prasanna

In recognition of the publication of the paper entitled

MobCurrent: YOLOBased People Flow Detection and Interactive Visual Analytics

Published In JETIR (www.jetir.org) ISSN UGC Approved (Journal No: 63975) & 7.95 Impact Factor

Published in Volume 12 Issue 4 , April-2025 | Date of Publication: 2025-04-05

Parin P
EDITOR

EDITOR IN CHIEF

ISSN 2349-5162

JETIR2504054 Research Paper Weblink http://www.jetir.org/view?paper=JETIR2504054 Registration ID : 558262

BIBLIOGRAPHY

BIBLIOGRAPHY

- ❖ YOLOv3: An Incremental Improvement – Real-time object detection model used in MobCurrent for fast and accurate people detection.
  https://arxiv.org/abs/1804.02767
- ❖ DeepSORT: Simple Online and Realtime Tracking with a Deep Association Metric – Used for maintaining object identity across frames.
  https://ieeexplore.ieee.org/document/8900533
- ❖ OpenCV: Open Source Computer Vision Library – Core for image processing, frame extraction, and object tracking.
  https://opencv.org/
- ❖ Flask: Web Development, One Drop at a Time – Lightweight web framework used for MobCurrent's interface.
  https://flask.palletsprojects.com/
- ❖ Ultralytics YOLOv8 – Advanced object detection model enhancing accuracy and speed.
  https://ultralytics.com/
- ❖ Matplotlib: Visualization with Python – Enabled dynamic plotting for trend and movement analysis.
  https://matplotlib.org/
- ❖ Pandas: Data Analysis and Manipulation Tool – Essential for managing processed data and generating analytical reports.
  https://pandas.pydata.org/
- ❖ CUDA Toolkit: GPU-Accelerated Computing for Deep Learning – Enabled GPU support to boost video processing efficiency.
  https://developer.nvidia.com/cuda-toolkit
- ❖ Plotly: Interactive Visualizations for Python – Enhanced the web interface with real-time interactive data dashboards.
  https://plotly.com/