AI-DeepVision: Crowd Monitoring & Density Map Generation

By: Ishita Deshpande

Table of Contents

Project Description

AI-DeepVision is a research-oriented project aimed at automating crowd monitoring using computer vision and deep learning techniques. The system preprocesses the ShanghaiTech dataset to generate density maps that represent crowd distribution across images, which are essential inputs for modern crowd counting models. The project includes image preprocessing, density map generation, dataset visualization, and modular PyTorch data pipeline development. It prepares high-quality input data for models like CSRNet, MCNN, SANet, and CANNet.

Introduction To Model Deployment

Model deployment is the final and most critical phase in the lifecycle of a deep learning system, where a trained and tested model is transformed into a usable, interactive, and operational application. While previous milestones focus on training accuracy and inference validation, deployment determines whether the system can function reliably in real-world conditions and deliver actionable insights to end users. In crowd monitoring applications, deployment is particularly important because predictions may directly influence safety decisions and emergency responses.

In the AI-DeepVision project, deployment involves integrating trained crowd counting models into a user-facing interface capable of real-time video processing, visualization, and alert generation. This milestone ensures that the complete pipeline—from input acquisition to decision notification—operates as a cohesive system. Milestone 4 focuses on deploying the system using a web-based framework, enabling real-time interaction, visualization of density maps, and automated email alerts when crowd thresholds are exceeded.

Deployment Architecture Overview

The deployed crowd monitoring system follows a modular architecture that separates concerns between computation, visualization, and alerting. The major components include:

- A deep learning inference backend using trained CSRNet models
- A web-based frontend for user interaction
- A visualization layer for displaying crowd density heatmaps
- An alerting mechanism based on SMTP email communication

This architecture ensures scalability, maintainability, and ease of use. Each component operates independently while contributing to the overall system functionality.

Model Loading and Resource Management

During deployment, trained model weights are loaded into memory and configured for inference-only execution. The deployment environment dynamically selects the available computational device (CPU or GPU) to optimize performance. Efficient resource management is essential to ensure low latency and stable operation, especially in real-time applications.

Inference-Only Execution Mode

In the deployed system, models operate exclusively in inference mode, where gradient computation is disabled. This reduces memory usage, improves execution speed, and ensures prediction stability. Inference-only execution is a core requirement for continuous real-time monitoring systems.

Streamlit as Deployment Framework

Streamlit is a lightweight web application framework designed for deploying data-driven and machine learning applications. It enables rapid conversion of Python-based inference pipelines into interactive web interfaces without requiring complex frontend development.

From a theoretical standpoint, Streamlit serves as the presentation and interaction layer, allowing users to interact with the deployed model through a browser-based interface while the backend handles inference and processing.

Multi-Tab User Interface Design

The deployed application uses a tab-based interface to separate different operational modes of the system. This design improves usability and clarity by organizing functionality into logical sections.

The two primary tabs include:
- Webcam Monitoring
- Video Upload Monitoring

Each tab represents a distinct input modality while sharing the same inference and alerting backend. This modular UI design allows easy extension of additional input sources in the future.

Webcam Monitoring Tab

The Webcam Monitoring tab represents the real-time operational mode of the deployed crowd monitoring system. Its primary purpose is to enable continuous, live analysis of crowd density using a connected camera device. This mode closely reflects real-world surveillance scenarios such as monitoring public spaces, entry points, or event venues, where immediate feedback and responsiveness are essential.

From a deployment perspective, this tab validates whether the trained models and inference pipeline can function reliably under real-time constraints. Unlike offline analysis, webcam-based monitoring requires uninterrupted execution, low latency, and stable output generation, making it a critical component of the deployed application.

Continuous Frame Acquisition and Control Flow

In this mode, frames are captured continuously from the webcam and processed sequentially. The user interface provides explicit controls to start and stop the live feed, allowing the user to manage system execution safely and intentionally. This control flow ensures that system resources are allocated only when required and prevents unnecessary processing when monitoring is paused.

Theoretical importance lies in controlled execution and lifecycle management. Real-time systems must handle initiation, execution, and termination gracefully to avoid memory leaks, device lockups, or inconsistent states.

Real-Time Preprocessing and Inference Pipeline

Each captured frame undergoes preprocessing to ensure consistency with training data characteristics. This includes color space alignment, normalization, and tensor conversion. Once preprocessed, the frame is passed through the crowd counting inference pipeline.

From a theoretical standpoint, this pipeline demonstrates the application of forward-only inference in a real-time environment. The system must complete preprocessing, inference, and visualization within a short time window to maintain acceptable responsiveness. This tab therefore evaluates the computational efficiency of the deployed model and its suitability for continuous execution.

Dynamic Model Selection in Live Monitoring

An important feature of the Webcam Monitoring tab is dynamic model switching based on estimated crowd density. The system evaluates crowd count estimates in real time and selects the most appropriate trained model optimized for the current density range.

This adaptive approach improves robustness by leveraging model specialization. In live environments, crowd density can change rapidly, and dynamic model selection ensures that predictions remain accurate across varying conditions without requiring manual intervention.

Real-Time Visualization and Heatmap Overlay

The Webcam Monitoring tab provides visual feedback by overlaying predicted density heatmaps onto the live video feed. This visualization allows users to interpret both the numerical crowd count and the spatial distribution of people within the scene.

Theoretical significance lies in enhancing interpretability. Visual overlays allow users to assess whether model predictions align with observed crowd patterns, which is especially important in safety-critical applications. The blending of raw frames with density maps preserves scene context while highlighting areas of concern.

Alert Triggering and User Notification

The live monitoring mode continuously evaluates predicted crowd counts against predefined thresholds. When a critical threshold is exceeded, the system automatically triggers an email alert.

From a deployment perspective, this mechanism transforms the system from passive observation to active intervention. The integration of alerts ensures that stakeholders are informed promptly, enabling rapid response to potentially unsafe crowd conditions. Safeguards are implemented to prevent repeated alerts for the same event, improving reliability and usability.

Operational Significance of Webcam Monitoring Tab

Overall, the Webcam Monitoring tab demonstrates the system's readiness for real-time deployment. It validates the complete integration of live input acquisition, inference, visualization, and alerting, confirming that the deployed application can operate continuously in practical surveillance scenarios.

Video Upload Tab

The Video Upload Monitoring tab provides an offline analysis mode for processing pre-recorded video files. This mode is particularly useful for post-event analysis, controlled evaluation, and demonstration purposes. It allows users to upload video recordings and analyze crowd behavior over time without requiring a live camera feed.

From a theoretical standpoint, this mode extends system usability beyond real-time monitoring, enabling analysis of historical data and validation of model behavior under known conditions.

Video File Handling and Temporary Storage

Uploaded video files are temporarily stored and processed sequentially. This approach ensures compatibility with large video files and avoids direct dependency on file paths or persistent storage.

Theoretical importance lies in safe and isolated file handling. Temporary storage prevents data corruption and ensures that uploaded content is processed independently of system state, enhancing deployment robustness.

Sequential Frame Processing and Temporal Analysis

The video upload mode decodes the uploaded video into individual frames, which are processed in chronological order. Each frame undergoes preprocessing and inference, similar to the webcam mode.

This sequential processing allows the system to analyze how crowd density evolves over time. Temporal patterns such as gradual crowd buildup or sudden surges can be observed, making this mode valuable for analytical and validation purposes.

Visualization of Crowd Dynamics

As with live monitoring, predicted density heatmaps are overlaid on video frames to visualize crowd distribution. This visualization enables users to correlate changes in crowd count with visual scene content.

Theoretical significance lies in combining quantitative and qualitative analysis. Users can observe both numerical trends and spatial distribution, leading to deeper understanding of crowd behavior.

Alert Mechanism in Offline Analysis

Although video upload mode is primarily analytical, the alert mechanism remains active. Alerts are triggered when crowd counts exceed critical thresholds, demonstrating consistency in system behavior across operational modes.

This design ensures that alert logic is decoupled from input source and remains uniform across the deployed application.

Comparative Value of Video Upload Mode

The Video Upload Monitoring tab complements live monitoring by offering repeatable and controlled analysis. It is especially valuable for testing, debugging, validation, and demonstration, as the same video can be analyzed multiple times under identical conditions.

Role in Overall Deployment Strategy

Together with the Webcam Monitoring tab, the Video Upload Monitoring tab ensures comprehensive deployment coverage. The system supports both real-time surveillance and offline analysis, making it versatile and adaptable to diverse crowd monitoring requirements.

Multi-Model Working

Dynamic Model Switching Based on Crowd Density

An important deployment concept implemented in the system is **dynamic model selection**. Based on the estimated crowd count, the system switches between two trained models optimized for different crowd density ranges.

Theoretical justification for this approach lies in specialization: models trained on different crowd distributions may perform better within specific density ranges. Dynamic switching improves overall accuracy and robustness by selecting the most appropriate model for the current scenario.

Heatmap Overlay and Density Visualization

Heatmap overlays provide an intuitive visual representation of crowd density by mapping predicted density values to color intensities. High-density regions are highlighted using warmer colors, while sparse regions appear cooler.

From a theoretical standpoint, heatmaps enhance interpretability by allowing users to visually verify whether model predictions align with observed crowd patterns.

The density map is resized to match the original frame dimensions and blended with the original image. This overlay technique preserves contextual visual information while highlighting crowd concentration areas, making it suitable for surveillance and monitoring applications.

Alerting Mechanism

In real-world crowd monitoring applications, merely estimating crowd density is insufficient unless the system can actively respond to potentially unsafe conditions. An alerting mechanism enables the system to transition from passive observation to proactive intervention. In the context of the AI-DeepVision project, the alerting system is designed to notify responsible stakeholders when crowd density exceeds predefined safety thresholds, thereby enabling timely preventive action.

Alerting systems are a critical component of intelligent surveillance applications, particularly in scenarios such as public gatherings, transportation hubs, and emergency management. The integration of automated alerts ensures that decision-makers are informed immediately when abnormal or hazardous crowd conditions arise, reducing reliance on continuous manual monitoring.

Concept of Threshold-Based Alerting

Threshold-based alerting is a widely used mechanism in monitoring systems, where alerts are triggered when a measured parameter crosses a predefined limit. In crowd monitoring, this parameter is the estimated crowd count derived from density map regression.

The threshold represents a safety boundary that distinguishes normal conditions from potentially dangerous overcrowding. When the predicted crowd count exceeds this boundary, the system interprets the situation as requiring attention. Thresholds are selected based on contextual factors such as physical space constraints, safety regulations, and operational requirements.

This approach offers simplicity, interpretability, and reliability. It allows clear mapping between model output and alerting behavior, ensuring that alerts are predictable and explainable.

Role of Email Alerts in Deployed Systems

Email-based alerts are a practical and widely supported method for delivering notifications in deployed monitoring systems. Unlike visual alerts that require users to be actively observing the system interface, email alerts provide asynchronous communication, ensuring that critical notifications are delivered even when the user is not directly interacting with the application.

Email alerts are particularly suitable for:

- Long-duration monitoring
- Remote supervision
- Multi-stakeholder notification
- Archival and audit purposes

In the deployed system, email alerts complement on-screen indicators, creating a multi-layered notification strategy.

Overview of SMTP Protocol

SMTP (Simple Mail Transfer Protocol) is a standardized communication protocol used for sending email messages over the internet. It defines the rules for transmitting email from a client to a mail server and between mail servers. SMTP operates at the application layer of the TCP/IP model and is designed for reliable, text-based message delivery.

The SMTP communication process involves several stages:

1. Establishment of a connection between the client and the mail server
2. Authentication of the sender
3. Transmission of message metadata (sender, recipient, subject)
4. Transmission of message content
5. Server acknowledgment and connection termination

This structured communication ensures reliable delivery of email alerts.

Authentication and Security in SMTP-Based Systems

Modern SMTP systems require authentication to prevent misuse and unauthorized access. Authentication mechanisms verify the identity of the sender before allowing email transmission.

In deployed applications, app-specific passwords are often used instead of regular account passwords. These passwords provide restricted access limited to email sending functionality, reducing security risk. This approach aligns with best practices for secure deployment.

SMTP connections are typically encrypted using TLS (Transport Layer Security) to protect message content and credentials during transmission. Secure channels prevent interception and tampering, ensuring confidentiality and integrity of alert messages.

Integration of SMTP Alerting into the Crowd Monitoring Pipeline

The alerting system is integrated directly into the inference pipeline of the deployed application. After each inference cycle, the predicted crowd count is compared against predefined alert thresholds. If the threshold condition is met, the alerting module is invoked.

This tight integration ensures:

• Minimal delay between detection and notification
• Consistent alert behavior across input modes
• Automated response without human intervention

By embedding alert logic within the inference loop, the system ensures real-time responsiveness.

Alert Frequency Control and Redundancy Prevention

A key challenge in alerting systems is preventing alert flooding, where repeated alerts are sent for the same ongoing condition. Excessive alerts can overwhelm users and reduce the effectiveness of notifications.

To address this, the system includes alert frequency control logic. Once an alert is sent for a particular event, further alerts are temporarily suppressed until the condition resets. This ensures that alerts remain meaningful and actionable.
From a theoretical perspective, alert suppression improves usability and aligns with human factors engineering principles.

Reliability and Fault Tolerance in Alerting Systems

Alerting systems must be reliable and resilient to transient failures. Email delivery mechanisms inherently provide reliability through server acknowledgments and retry mechanisms. If a message cannot be delivered immediately, mail servers typically attempt redelivery, increasing the likelihood of successful notification.
In addition, decoupling alert logic from visualization ensures that alerts can still be generated even if the user interface experiences temporary issues.

Advantages of SMTP-Based Alerting in Crowd Monitoring

The use of SMTP-based email alerts offers several advantages:
• Platform independence
• Wide compatibility with email clients
• Asynchronous communication
• Auditability through email records
These characteristics make SMTP a suitable choice for safety-critical monitoring applications.

Role In System

The alerting system implemented in the AI-DeepVision project is a crucial component that elevates the application from a visualization tool to an intelligent decision-support system. By leveraging SMTP-based email notifications and threshold-driven logic, the system ensures timely and reliable communication of critical crowd conditions.
The integration of alerting with real-time inference and deployment infrastructure demonstrates a comprehensive and practical approach to crowd monitoring. This alerting mechanism enhances safety, usability, and operational effectiveness, making the deployed system suitable for real-world surveillance and crowd management applications.

Conclusion

Milestone 4 successfully transformed the AI-DeepVision project from a research-oriented prototype into a fully deployed crowd monitoring application. Through integration of a web-based interface, real-time inference, dynamic model selection, heatmap visualization, and automated alerting, the system demonstrates practical usability and operational readiness. This milestone confirms that the trained models can function reliably within an end-to-end deployment pipeline.

Overall Project Conclusion

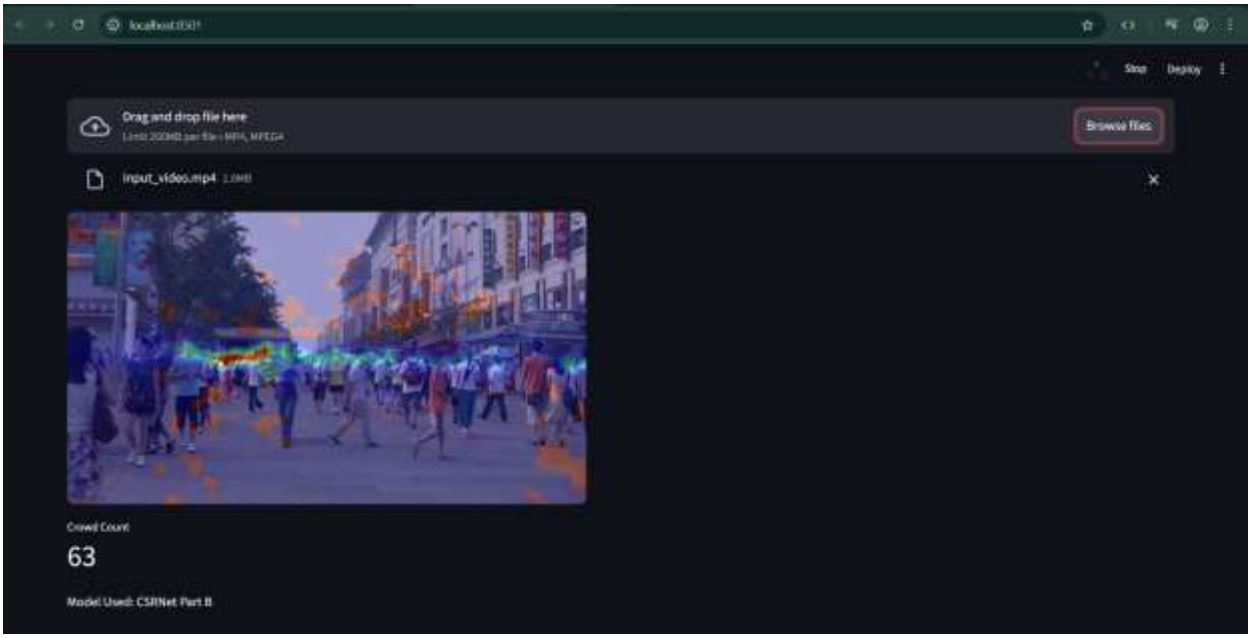
The AI-DeepVision project presents a complete end-to-end crowd monitoring solution built using deep learning and computer vision techniques. Beginning with dataset preprocessing and density map generation, the project systematically progressed through model training, evaluation, rigorous testing, and final deployment. Each milestone addressed a critical stage in the system lifecycle, ensuring technical correctness, robustness, and real-world applicability.

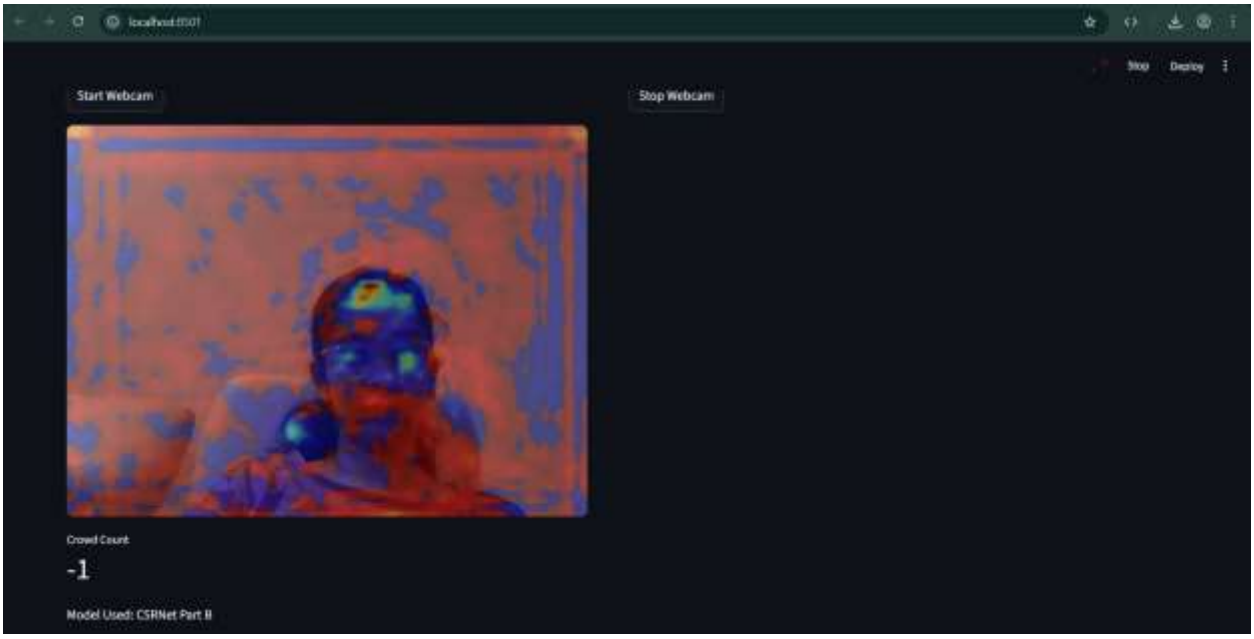
By adopting density-based regression models, the system effectively handles challenges associated with high-density crowds, occlusion, and perspective distortion. Comprehensive testing across webcam feeds, video files, and frame-wise inputs validated model generalization and stability. The final deployment phase integrated visualization and alert mechanisms, transforming the system into an actionable crowd monitoring platform.

Overall, the project demonstrates a strong alignment between theoretical foundations and practical implementation, resulting in a scalable, reliable, and deployable crowd monitoring system suitable for real-world safety and surveillance applications.

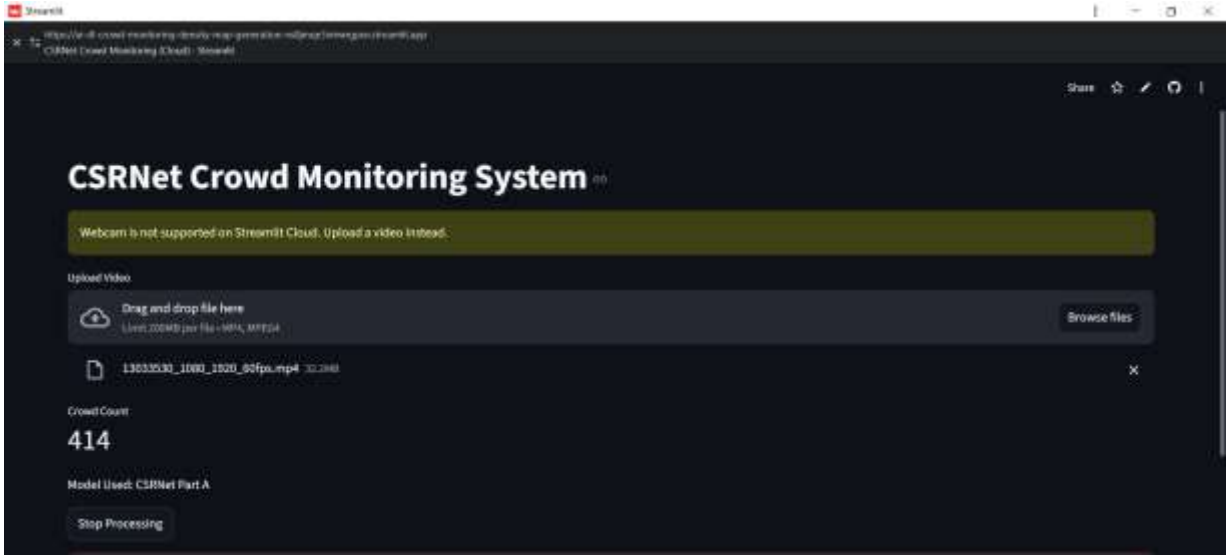
Output

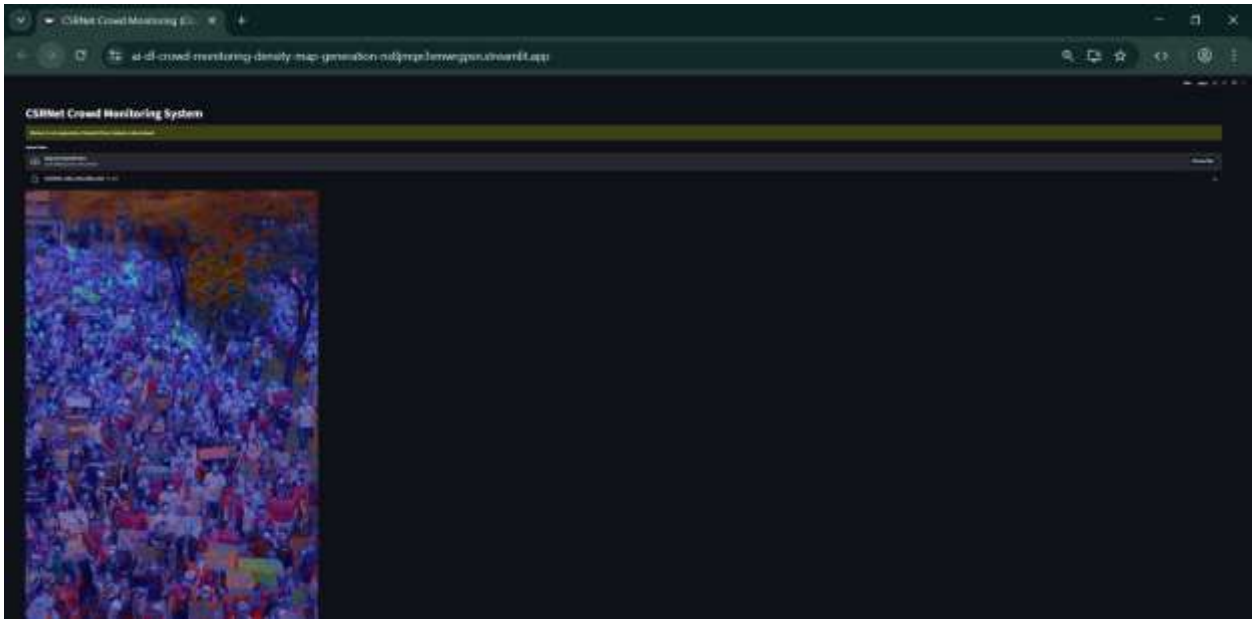
1. **Local Host**





Streamlit Deployed Application





Alert Message (Crowd > 120)