DeepVision Crowd Monitor: AI for Density Estimation and Overcrowding Detection

Sreeja Banoth

1. Project Description

The **DeepVision Crowd Monitor** is a Python-based deep learning application designed to estimate crowd density and detect overcrowding in images, videos, and live webcam feeds. The system leverages a CSRNet-based convolutional neural network to generate density maps and estimate the total number of people present in a scene.

The primary objective of this project is to demonstrate how computer vision and deep learning techniques can assist in crowd monitoring for public safety, event management, surveillance, and smart city applications. A web-based dashboard built using **Streamlit** enables interactive visualization of results in real time.

2. Dataset Used

For model development and reference, the system is designed to work with datasets commonly used in crowd counting research, such as the **ShanghaiTech Crowd Counting Dataset**.

Dataset characteristics:

- Crowd images containing sparse to highly dense scenes
- Ground truth annotations representing head locations
- Suitable for training density-estimation models like CSRNet

In this implementation, the dashboard focuses on **inference and visualization**, while training support is provided through a separate script.

3. Environment Setup

The project is implemented using **Python** and executed locally on a CPU-based system.

Tools and Libraries Used:

- **Python 3.10**
- **PyTorch** – deep learning framework
- **OpenCV** – image and video processing
- **NumPy** – numerical computation
- **Matplotlib** – density map visualization
- **Streamlit** – web-based dashboard interface

A local virtual environment can optionally be used to manage dependencies.

4. Data Preprocessing

Before inference, input images and video frames undergo preprocessing to ensure compatibility with the CSRNet model.

4.1 Image Loading

- Images are loaded using OpenCV in BGR format.
- Converted to RGB for visualization and processing consistency.

4.2 Resizing and Normalization

- Images are resized to match model expectations.
- Pixel values are normalized to improve numerical stability.

4.3 Tensor Conversion

- Images are converted into PyTorch tensors.
- Tensor shape is rearranged to $(C \times H \times W)$ and moved to the selected device (CPU/GPU).

5. Proposed System

The overall system workflow consists of the following stages:

1. **Input Acquisition**
    - Image upload
    - Video upload (short clips)
    - Live webcam feed
2. **Preprocessing**
    - Resize
    - Normalize
    - Tensor conversion
3. **CSRNet Inference**
    - Feature extraction
    - Density map prediction
4. **Crowd Count Estimation**
    - Density map values are summed to estimate total crowd count

5. **Overcrowding Detection**
    - Estimated count is compared with a user-defined threshold
    - Visual alert is displayed if the threshold is exceeded

6. CSRNet Model Overview

CSRNet is a convolutional neural network architecture specifically designed for crowd counting tasks.

Key characteristics:

- Front-end convolutional layers for feature extraction
- Dilated convolution back-end to capture wider contextual information
- Output is a density map instead of direct count regression

The predicted density map enables spatial visualization of crowded regions and provides accurate crowd estimation by integration.

7. Training Pipeline

The project includes a training script (train_csrnet.py) that supports:

- Loading preprocessed crowd datasets
- Training CSRNet on density maps
- Saving model checkpoints
- Evaluating performance using error metrics

In the current dashboard deployment, **pretrained weights are optional**, and the system can run in demo mode without trained weights.

8. Image-Based Crowd Density Estimation

Using the Streamlit dashboard:

- Users can upload a single crowd image
- The system predicts a density map
- The total crowd count is displayed
- Overcrowding alerts are shown when thresholds are exceeded

This feature provides a fast and intuitive way to analyze crowd density in static images.

9. Video-Based Crowd Density Estimation

The system supports crowd estimation on **short pre-recorded videos**.

Workflow:

1. Video is uploaded via the dashboard
2. Frames are extracted using OpenCV
3. Selected frames are passed through CSRNet
4. Average crowd count is computed
5. Alert status is displayed

Due to CPU-based processing, short videos (5–8 seconds) are recommended for smooth performance.

10. Real-Time Crowd Monitoring Using Webcam

A live webcam monitoring feature is integrated into the dashboard.

Functionality:

- Captures frames from the local webcam
- Displays live video feed in the browser
- Demonstrates real-time monitoring capability

This module serves as a proof-of-concept for real-time deployment and can be extended with optimized inference pipelines.

11. Web-Based Dashboard (Streamlit)

The Streamlit-based dashboard acts as the user interface for the system.

Features:

- Image upload
- Video upload
- Live webcam feed
- Density map visualization
- Crowd count display

- Overcrowding alert system
- User-configurable threshold control

The dashboard enables easy interaction without requiring command-line usage.

12. Alert System

The current implementation includes:

- **Visual alerts** displayed directly on the dashboard
- Threshold-based overcrowding detection

Email and SMS alerts are considered **future enhancements** and can be integrated using SMTP services.

13. Deployment

The application is deployed **locally** using Streamlit.

- Run command: streamlit run app_streamlit.py
- Accessed via: http://localhost:8501

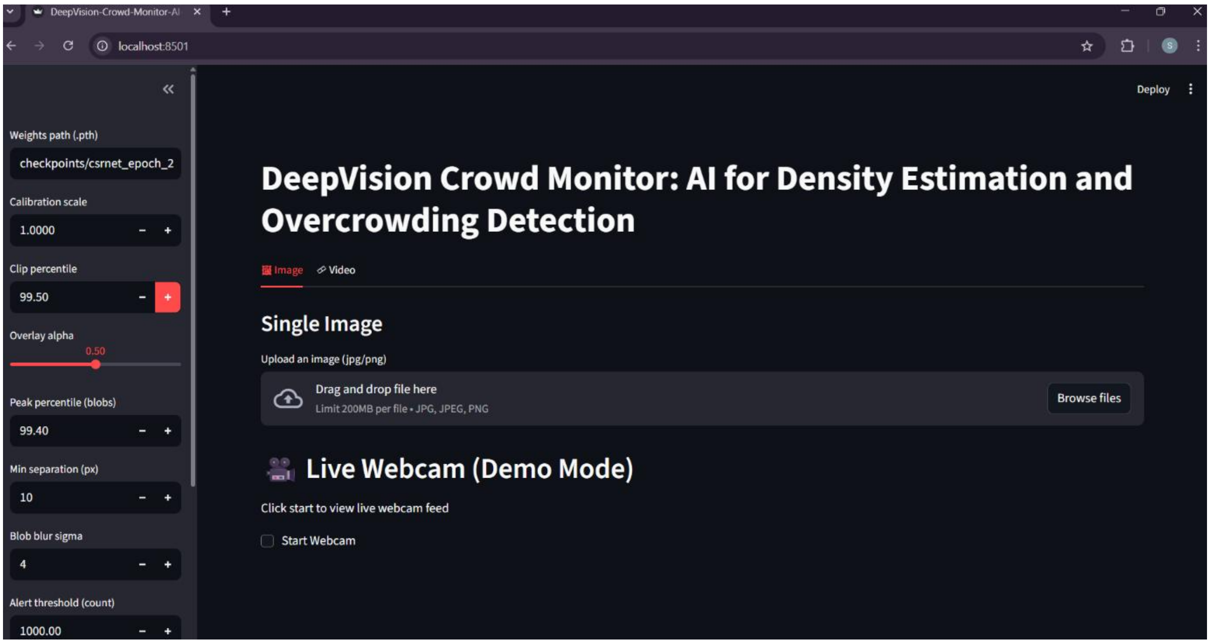
This approach allows rapid testing, demonstrations, and academic evaluation.

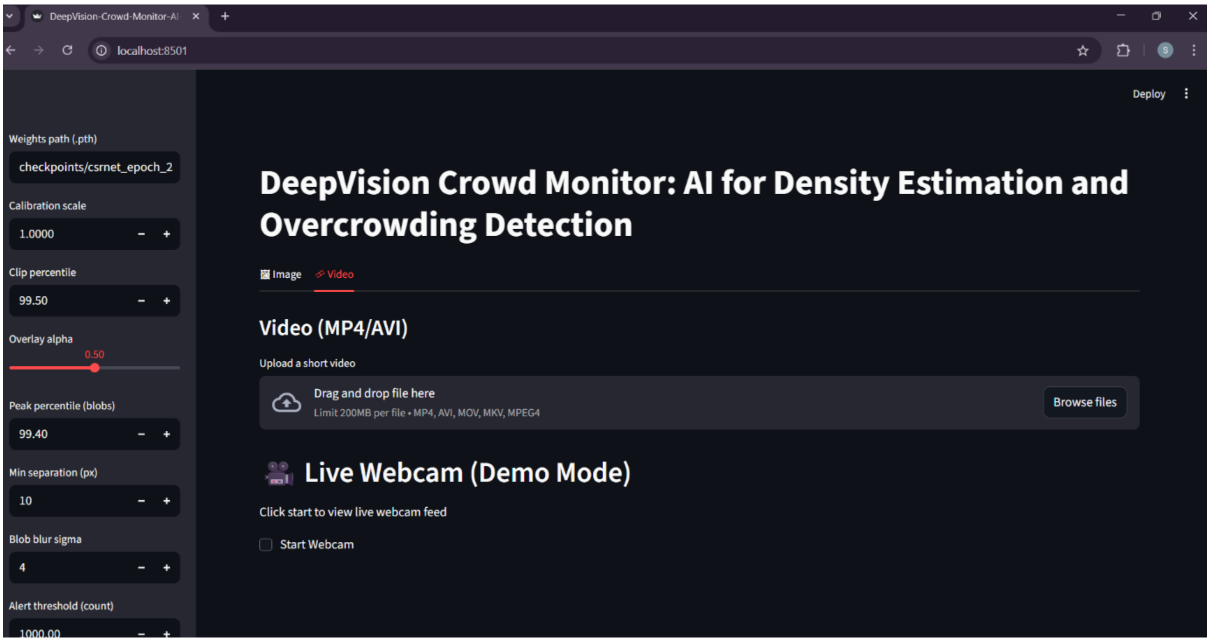
14. Conclusion

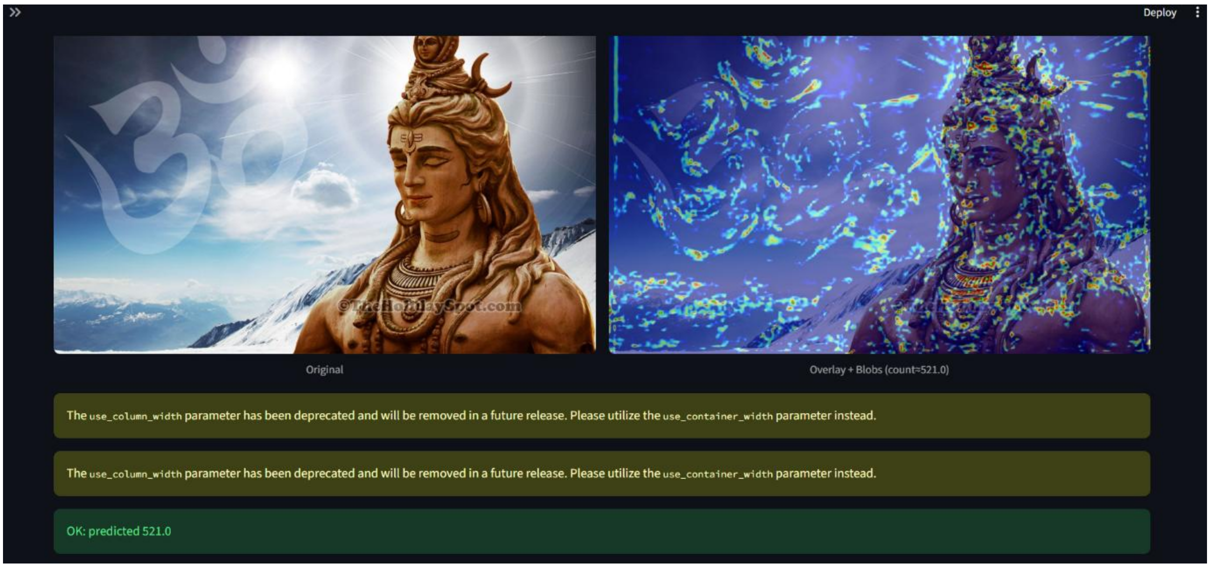
The DeepVision Crowd Monitor successfully demonstrates how deep learning-based density estimation can be applied to real-world crowd monitoring scenarios. By integrating CSRNet with a Streamlit dashboard, the system provides an interactive platform for image, video, and webcam-based crowd analysis.

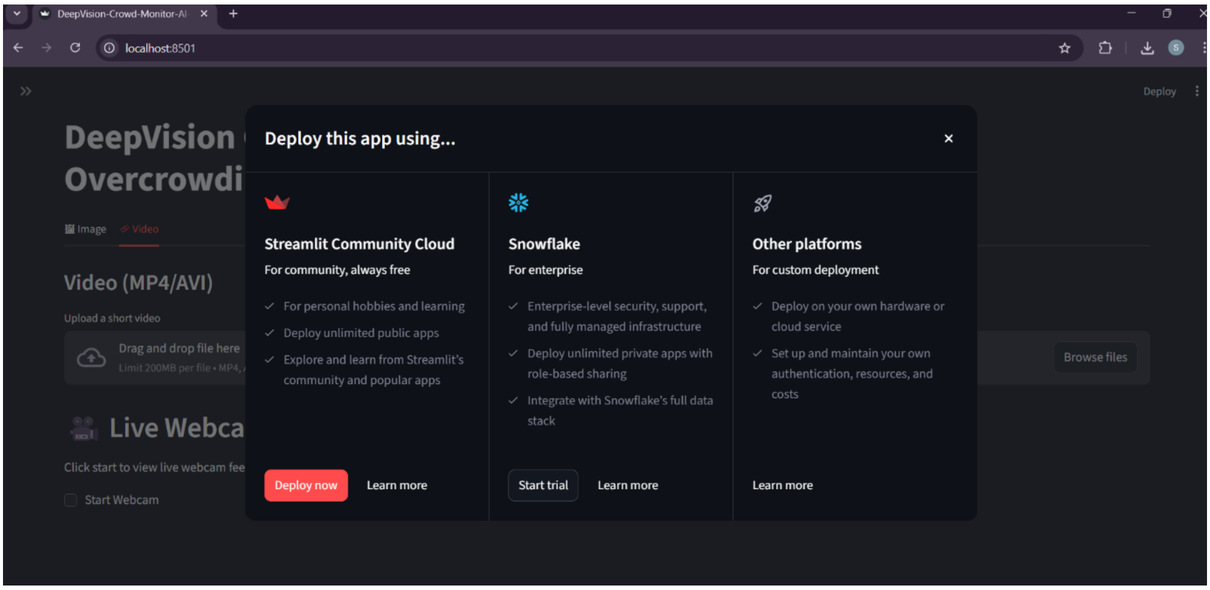
The project highlights the practical challenges of real-time video processing and emphasizes the importance of efficient preprocessing and inference strategies. Overall, the system serves as a strong foundation for intelligent surveillance and crowd management applications, with scope for future enhancements such as optimized real-time inference and automated alert notifications.

Output Screenshots






Deploy
©TheHolidaySpot.com
Original
Overlay + Blobs (count=521.0)
The use_column_width parameter has been deprecated and will be removed in a future release. Please utilize the use_container_width parameter instead.
The use_column_width parameter has been deprecated and will be removed in a future release. Please utilize the use_container_width parameter instead.
OK: predicted 521.0


DeepVision-Crowd-Monitor-AI
localhost:8501
Deploy
DeepVision
Overcrowdi
Image
Video
Video (MP4/AVI)
Upload a short video
Drag and drop file here
Browse files
Live Webca
Click start to view live webcam fee
Start Webcam
Deploy this app using...
Streamlit Community Cloud
For community, always free
For personal hobbies and learning
Deploy unlimited public apps
Explore and learn from Streamlit's community and popular apps
Deploy now
Learn more
Snowflake
For enterprise
Enterprise-level security, support, and fully managed infrastructure
Deploy unlimited private apps with role-based sharing
Integrate with Snowflake's full data stack
Start trial
Learn more
Other platforms
For custom deployment
Deploy on your own hardware or cloud service
Set up and maintain your own authentication, resources, and costs
Learn more