

DeepVision Crowd Monitor: Real-Time Density and Safety Analysis

Kushul Aleti

December 2, 2025

Abstract

A state-of-the-art deep learning system for high-precision crowd counting and density map generation, designed to enhance public safety, optimize resource allocation, and detect overcrowding in real-time environments. This project implements the **CSRNet** architecture for density estimation and integrates **YOLOv8** for real-time object detection and bounding box visualization in video streams.

1 Key Features

- **High-Precision Counting:** Utilizes CSRNet to provide accurate crowd count estimates in congested and occluded scenes, exceeding the performance of traditional object detection methods.
- **Density Map Visualization:** Generates visual heatmaps that instantly highlight areas of high and low density, crucial for quick operational awareness.
- **Real-Time Alerts:** Implements a configurable threshold system to trigger immediate visual and audio alerts when a crowd count exceeds safe limits (e.g., in the video stream mode).
- **Dual Mode Operation:** Supports both static image analysis and live video stream processing (file upload or local webcam integration).
- **Professional Streamlit UI:** Provides an intuitive, web-based interface for easy interaction and display of results.

2 Core Technology: CSRNet

The foundation of the counting mechanism is the **CSRNet (Context-aware Scale-recurrent Network)** model, which excels in congested scenes by focusing on density rather than individual bounding box detection.

The estimation process involves:

1. **Feature Extraction:** Using the VGG-16 architecture to extract high-level visual features.
2. **Density Mapping:** Employing a Dilated Convolutional Neural Network (D-CNN) back-end to generate a high-resolution density map. The sum of all pixel values in this map yields the estimated crowd count.

3 Project Structure

The repository contains the following main operational components:

File	Description
CSRNet.py	Main Application File. Contains the Streamlit UI logic, handles I/O, integrates model loading.
model.py	Contains the definition of the CSRNet class, including the VGG front-end and the Dilated CNN back-end.
weights.pth	The pre-trained weights file for the CSRNet model. (Required for execution).
yolov8n.pt	The lightweight pre-trained weights for the YOLOv8 object detection model (used for bounding box detection).

4 Setup and Installation

Follow these steps to get the DeepVision Crowd Monitor running on your local machine.

4.1 Prerequisites

- Python 3.8+
- NVIDIA GPU (Recommended for faster processing via CUDA/PyTorch)

4.2 Clone the Repository

```
git clone https://github.com/stkwer/DeepVision-Crowd-Monitor-AI-for-Density-Estimation-and-Overcrowding-Detection.git  
cd DeepVision-Crowd-Monitor-AI-for-Density-Estimation-and-Overcrowding-Detection
```

4.3 Set up the Environment

Create a virtual environment and install the necessary dependencies:

```
# Create and activate environment (Linux/macOS)  
python3 -m venv venv  
source venv/bin/activate  
# Install dependencies  
pip install -r requirements.txt
```

(Note: You will need to create a `requirements.txt` file listing `streamlit`, `torch`, `torchvision`, `numpy`, `opencv-python`, and `ultralytics`.)

4.4 Obtain Model Weights

Ensure you have the necessary pre-trained weights:

- Place the **CSRNet weights** file, named `weights.pth`, in the root directory.
- Ensure the **YOLOv8 weights** file, `yolov8n.pt`, is available.

4.5 Run the Application

Start the Streamlit server from your terminal:

```
streamlit run CSRNet.py
```

The application will open automatically in your default web browser (usually at <http://localhost:8501>).

5 Configuration

The sidebar in the Streamlit application allows you to configure the **Alert Frequency Threshold**. This is the critical crowd count level (e.g., 30 people) that triggers a high-priority alert in the video processing mode, essential for proactive safety management.