CrowdSense
Real-time crowd formation & risk forecasting

Introduction

This project explores a two-stage approach to understanding and managing crowds from CCTV-like video. First, we detect and quantify human gatherings indirectly, using a state-of-the-art density map model to produce head points, counts, and zone-level crowd signals, rather than relying on fragile person detection. Second, we use the temporal evolution of these signals - density, growth, compaction, and flow near bottlenecks- to estimate whether an existing gathering is drifting toward hazardous conditions in the near future. Together, these stages aim to turn raw video streams into interpretable early-warning signals that can help operators spot dangerous crowd build-up before it turns into an incident.

Research Question

Can we detect when a crowd gathering forms in CCTV video and, from its early visual patterns over time, forecast whether it will become high-risk (for example, risk of crowd crush due to overcrowding and violence) within the next X minutes

Literature Review and Existing Approaches

Inspiration and Background

Detecting human gatherings in public spaces is a crucial capability for urban safety, public health, and event monitoring, particularly in dense city environments. Traditional methods for identifying gatherings often rely on explicit detection and tracking of individuals, a task made difficult in real-world scenarios where people are partially occluded, appear at varying scales, or blend into complex urban backdrops. To overcome these challenges, modern computer vision has shifted toward **density-based crowd analysis**, which treats people not as discrete objects to detect but as contributors to a continuous density distribution over the scene [2]. This perspective allows us to estimate how many people are present and where they are concentrated, even when visual conditions are challenging, and is exemplified by recent density map models such as CSRNet [2] and DM-Count [1] that produce high-quality density fields in congested scenes.

However, in many tragic crowd incidents, such as crushes at festivals, stadiums, or religious events, the core problem is not just that a gathering exists, but that a normal crowd silently evolves into a dangerous one without timely warning [3]. Once we can detect and quantify gatherings in the scene, the next challenge is to understand how those gatherings change over time and whether they are drifting toward hazardous conditions. Instead of relying only on static snapshots, **risk escalation forecasting** looks at how density, movement, and compaction evolve - whether people are packing closer together, slowing down near bottlenecks, or being unable to escape the area, following insights from empirical studies of crowd disasters and congestion metrics that jointly consider density and flow [3,4]. By turning these temporal patterns into an early warning signal, the system aims to forecast danger a few minutes in advance, giving operators a chance to intervene before a dense crowd crosses the line into a hazardous situation, in line with recent work that uses learned crowd features (e.g., density and severity levels) to predict critical states and anomalies in advance [5].

Positioning of This Research

This project builds on advances in crowd counting via **density map regression**, a line of research that bypasses hard detection in favor of learning a smooth, continuous representation of human presence in each frame [2]. Inspired by this perspective, the work adopts **DM-Count** (Distribution Matching for Crowd Counting), a state-of-the-art density map model that offers both high accuracy and well-localized peaks without requiring explicit head bounding boxes during training [1]. By emphasizing **distribution matching** rather than pixel-wise similarity, DM-Count learns from sparse point annotations and generates precise, interpretable density maps [1]. These maps serve as a proxy for physical presence, from which head positions and, subsequently, gatherings can be inferred through spatial aggregation over time, building on a broader family of density-based models such as CSRNet that have demonstrated high-quality density estimation in congested scenes [2]. This density-driven approach is particularly aligned with smart city surveillance goals, where dynamic crowd behavior and limited per-person visibility demand robust, indirect estimation techniques.

On top of this density-based representation, the project extends its scope from *detecting and* ***quantifying*** gatherings to ***forecasting*** when they may become hazardous. Rather than treating each frame independently, the system uses the time series of density-derived signals, such as people per square meter, density growth rate near bottlenecks, compaction of clusters, and indicators of flow or stagnation, guided by decades of crowd safety research on critical density, stop-and-go waves, and turbulence in large gatherings [3]. Video-based congestion measures that explicitly couple density with motion, such as the **crowd congestion level** introduced for risk assessment in surveillance footage, further motivate the use of joint density-flow indicators for risk modeling [4]. Recent work on simulated crowd data shows that these types of features can drive both **unsupervised anomaly detectors** and **supervised sequence models** for early warning of hazardous states, by learning how high density, low speed, and directional conflicts precede dangerous crowd conditions [5]. In this way, the project is positioned at the intersection of modern computer vision and crowd safety: it leverages state-of-the-art density maps to obtain reliable, geometry-aware crowd measurements, and then applies risk-escalation modeling to

transform those measurements into **interpretable early-warning signals** to support real-time decision-making in urban environments.

Prior Work Relevant to Our Approach

Density-Based Crowd Counting

Modern crowd counting predominantly relies on density map regression, where CNNs learn to predict a continuous density field instead of explicit detections. CSRNet established a strong baseline for highly congested scenes by showing that dilated convolutions can produce accurate density estimates in complex urban imagery [2]. DM-Count advanced this line by framing learning as distribution matching between predicted density and point annotations, yielding sharper, better-localized maps and state-of-the-art count accuracy across several benchmarks [1]. Our work adopts this density-regression paradigm and specifically builds on DM-Count's ability to produce high-quality density fields that are suitable not only for counting but also for extracting head positions and zone-level signals.

From Density Maps to Gatherings

Several density-based methods implicitly support localization and hotspot reasoning by treating density peaks and local maxima as proxies for individual heads or crowd clusters [1,2]. This enables downstream tasks such as identifying crowded subregions or estimating per-region occupancy, even when explicit detection is unreliable. Our approach formalizes this idea into a simple, reusable interface: per-frame head points, total count, and zone-level counts, which together provide an operational definition of "gatherings" without introducing a separate gathering detection network.

Crowd Risk, Congestion, and Safety Indicators

Research on crowd disasters and safety has highlighted how dangerous conditions emerge when high density combines with restricted movement and geometric constraints. Helbing et al. showed that extreme densities in confined geometries can lead to stop-and-go waves and turbulent crowd motion, which often precede crush incidents [3]. Bek and Monari proposed a video-based "crowd congestion level" that explicitly couples local density with reduced motion to quantify congestion risk in surveillance footage [4]. These works motivate our choice of risk indicators, density per area, growth near bottlenecks, compaction, and low flow, as physically meaningful signals derived from video.

Learning-Based Crowd Anomaly and Risk Prediction

Recent work has explored learning-based models that operate on crowd-level features to detect or predict hazardous conditions. The SIMCD framework demonstrates how simulated crowd data with labels for "severity" and abnormal states can be used to train both **unsupervised anomaly detectors** (e.g., one-class models) and **supervised sequence models** (e.g., LSTMs) for early warning of dangerous crowd configurations [5]. This supports the idea that short

histories of features such as density, speed, and directional consistency are sufficient to forecast risk escalation. In our setting, the same density-derived, geometry-aware features can feed **unsupervised models** like **one-class SVM** (and related anomaly detectors) to learn the envelope of normal crowd behavior from non-incident periods, alongside **rule-based baselines** and, where appropriate, supervised temporal models. Across all these variants, we emphasize keeping the resulting risk scores and their drivers interpretable, so that alerts can be traced back to concrete signals such as rising density, low flow, or unusual motion patterns.

Our Approach: Key Differences

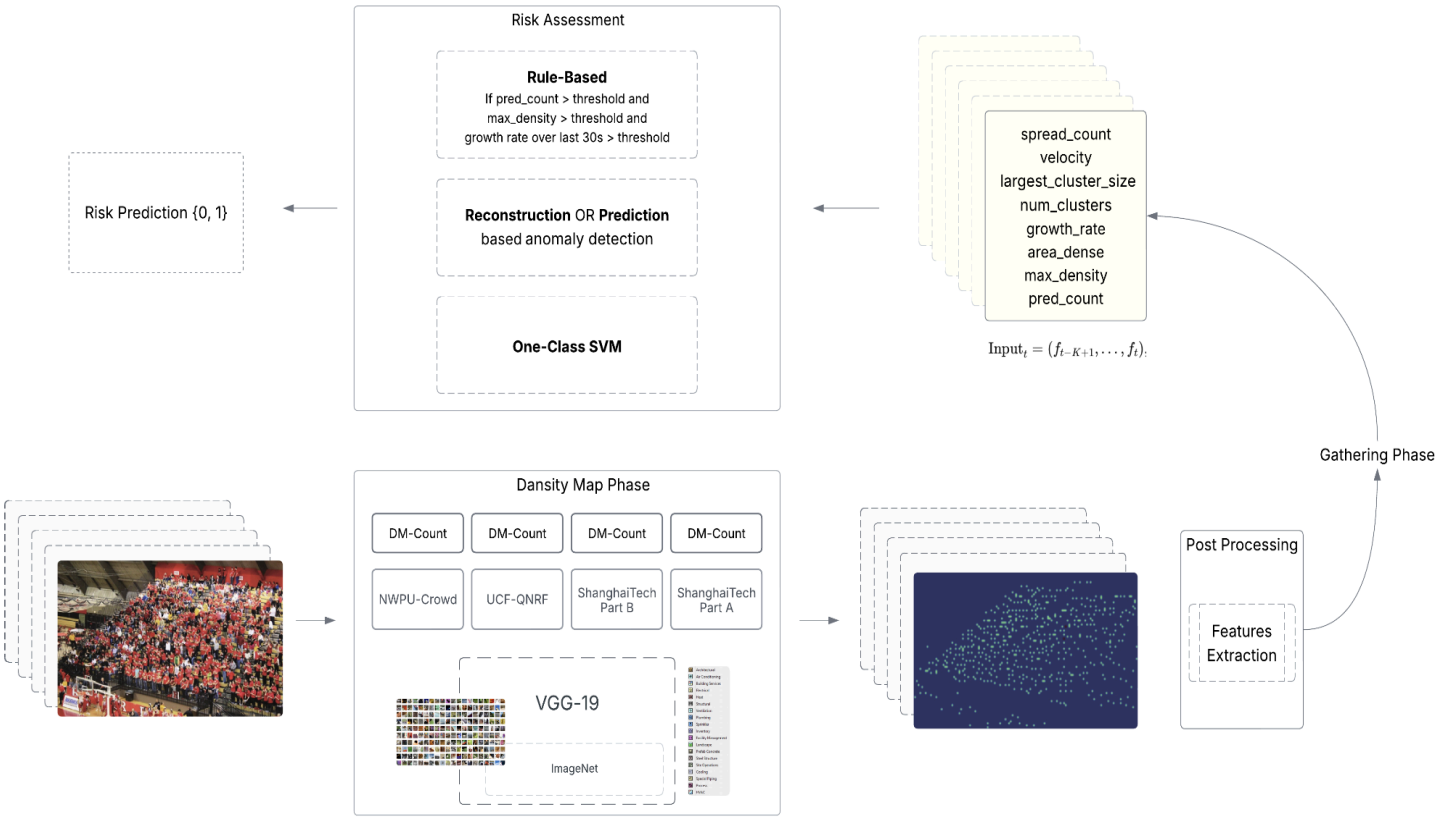
In summary, prior work has shown how to:

- obtain reliable density maps from single frames [1,2]
- identify critical density-flow patterns associated with disasters [3,4]
- use crowd-level features to detect or predict anomalous crowd states [5].

Our contribution is to connect these strands into a two-layer architecture: first, a state-of-the-art density model producing standardized, geometry-aware gathering signals; second, a risk escalation layer that transforms these signals into zone-level risk scores and human-readable drivers. This positions the system as a bridge between advanced crowd counting models and operational, interpretable early-warning tools for urban surveillance.

Methodology

High-Level Architecture



CrowdSense Pipeline Steps

The CrowdSense follows an end-to-end Sense-Detect-Predict pipeline to address our research question: “Can we detect when a crowd gathering forms in pictures and from its early visual patterns, forecast whether it will become high risk within the next few minutes?”

1. Overall Pipeline (Sense → Detect → Predict)

- Goal: From pictures, detect when a crowd gathering forms and forecast within the next few minutes whether it will become high-risk.
- Two-layer architecture:
  Layer 1: Computer vision → geometry-aware crowd signals (counts, density, head positions).
  Layer 2: Temporal risk modeling → early warning about escalation.

2. Sense Phase (Video Ingestion & Zones)

- Input: Static images. Zones: Scene divided into sub-regions (e.g., exit area, corridor, sections of plaza).
  1. Each zone is cropped/masked, resized, and normalized.
  2. Output of Sense: per-zone frames per time step, ready for the model.

3. Detect Phase (Model A: DM-Count)

The Detect phase is where CrowdSense converts each zone frame into crowd presence signals using a density-based crowd counting model.

For every zone, for every sampled frame, the system should output:

1. A density map — a 2D heatmap showing where people are.
2. A total headcount — the sum of the density map values.

This step becomes the foundation for detecting crowd gatherings (Step 5) and generating risk signals (Step 6).

- Uses DM-Count, a pretrained density-based crowd counting CNN.
- For each zone frame:
  1. Outputs a density map (2D field of “people density”).
  2. Total count = sum (integral) of density map.
- No on-site training; relies on generalization of pretrained weights.
- Output per zone/frame:
  1. Density map image.
  2. Numeric crowd count.

Why use Density-Based Counting (Instead of Person Detectors)? Traditional person detectors (YOLO, Faster R-CNN etc.) fail in:

- very crowded scenes
- occlusions
- overhead image angles
- small/blurred people

Density-based models bypass explicit bounding boxes and directly predict a smooth "density heatmap" where peaks correspond to heads. This is far more robust in crowd-dense environments.

The paper adopts DM-Count (Distribution Matching for Crowd Counting) as the Model A.

What DM-Count Does?

- Takes an image (zone frame) of people.
- Outputs a density map.
- The integral of the density map gives the crowd count.
- Produces smooth, well-localized peaks that align with head positions.

Why DM-Count Was Chosen

- State-of-the-art count accuracy on ShanghaiTech, UCF-QNRF, JHU.
- Produces clean, localizable peaks → useful for head point extraction.
- Requires no per-person detection.
- Robust in high-density image scenes.

How It Is Used in CrowdSense

- Pretrained weights are used (no retraining).
- Frame is resized & normalized to match the DM-Count input spec.
- Inference returns:
  - a 2D density map
  - a scalar count
- No domain-specific fine-tuning is required.

4. From Density Maps to Head Points & Per-Frame Signals

This section transforms continuous density maps into **discrete head positions and crowd metrics**, creating a spatially meaningful snapshot that can be tracked from frame to frame and used to detect gatherings and evaluate risk.

Why is this step important?

This is the standardization layer between raw vision output and higher-level reasoning.

It provides:

- A compact, interpretable representation of the crowd
- Reduction of raw image complexity
- A geometry-aware snapshot that can be tracked over time
- The spatial foundation for clustering, density, compaction, and stagnation detection

Without this step:

- You cannot detect gatherings (which requires knowing "who is close to whom").
- You cannot measure compaction or dispersion.
- You cannot model flow or stagnation.

- The prediction model would lack meaningful signals.

CrowdSense converts the raw density map output from DM-Count (Model A) into more interpretable, geometry-aware signals that can be used for gathering detection (Section 5) and risk prediction (Section 6).
This section is essentially the bridge between computer vision and crowd modeling.

What is a Density Map?
After Step 3 (detect phase), for each zone frame, we have:

- A density map: a 2D array where each pixel value ≈ represents the fractional number of people at that spot.
- A total headcount: the sum of the density map.

Example: If a particular pixel has a value of 0.3, it means "0.3 of a person is likely located here," and summing all pixels gives an accurate population count.
But density maps are continuous fields. For gathering detection, we need discrete head locations. Section 4 performs this conversion.
Steps:

- Convert the density map into:
    - Head points: local maxima in the density map (above a threshold) → (x, y) head locations.
    - Summary stats: total count, average density, max local density, etc.
    - If calibrated: density in persons/m².
- These form a geometry-aware snapshot per zone per frame:
    - Where people are.
    - How many.
    - How packed they are.
- Raw images don't need to be stored; only these structured signals.

Extracting Head Points From Density Peaks:
The idea: Treat the local maxima (peaks) in the density map as the approximate positions of individual heads.
This will work because:

- DM-Count produces smooth density maps with strong, well-localized peaks.
- In crowded scenes, bounding-box detectors fail, but density peaks remain robust.

Procedure

1. Scan the density map for local maxima.
2. Apply a minimum density threshold to avoid noise.
3. Each local maximum (peak) → a predicted head location.
4. Convert pixel coordinates → (x, y) head points within the zone frame.

Output of this step: a list of points like: [(x1, y1), (x2, y2), ..., (xK, yK)], where K = the crowd count estimate

Motivation of converting density maps to head points:
Head points give:

- Discrete individuals → needed for clustering (Section 3.5).
- Spatial structure → needed for risk signals (compaction, dispersion).
- Geometry-aware analysis → density by area, convex hull, spacing.

Density maps alone cannot easily tell:

- who is near whom,
- whether a compact cluster exists,
- whether a gathering persists over time.

Head points solve this.

Generating Per-Frame Crowd Metrics:
Besides the head points, Section 4 extracts several per-frame crowd descriptors:
A. Total Count: Already computed from the sum of density map.
B. Average Density

- avg density = total count / zone area
- If calibration exists → people per square meter.
  If not → pixels per pixel² (relative density).

C. Maximum Local Density
The highest value in the density map—useful for spotting very packed spots.
D. Spatial Features - these use the head point coordinates:

- spread/dispersion
- cluster radius
- convex hull area
- nearest-neighbor distances

These metrics become the raw ingredients for identifying crowd gatherings (Section 5) and computing risk signals (Section 6).

5. Gathering Detection From Head Points Over Space and Time

This Section explains how CrowdSense determines when a meaningful crowd gathering has formed — not just a few people standing together for a moment, but a true gathering that persists and may evolve into a dangerous crowd.

Problem being solved:
CrowdSense must decide:

- When does a crowd gathering actually exist?
- Is this cluster meaningful, or just people passing by?
- Has it existed long enough to monitor?
- How does it evolve as time passes?

To do this, Section 5 creates clusters of people and tracks them across frames.

This section turns head point coordinates (from Section 4) into gatherings, which are clusters of people that:

- are close together (spatial condition),
- have a minimum number of people (size condition),
- and persist long enough to matter (temporal condition).

Formal Definition of a Gathering: a gathering is at least N people within radius R, persisting for at least T seconds/frames.
A gathering is defined by three parameters: (parameter, meaning)

- N - Minimum number of people required to call it a gathering (e.g. 5 or 10)
- R - Proximity radius—how close individuals must be to be considered part of the same crowd (in meters or pixel distance).
- T - Minimum time duration the cluster must persist (e.g., 10 seconds).

A gathering exists only if all three conditions are met.

Definition: a gathering is at least N people within radius R, persisting for at least T seconds/frames.

- Steps:
    - Spatial clustering each frame (e.g., DBSCAN):
        - Points within distance R → clusters.
        - Discard clusters with < N people.
    - Temporal consistency:
        - Track clusters across frames by proximity/overlap.
        - If a cluster meets size ≥ N and persists ≥ T → confirmed gathering.
        - If it dissolves earlier → transient, ignored or timer restarts.
- For each gathering, maintain a time series:
    - Size, position, area, etc.
    - Multiple gatherings can be tracked simultaneously.

Detailed Steps:

Step 1 - Spatial Grouping (clustering at each image)

At each time step (each sampled frame):

1. We take the head point coordinates extracted earlier.
2. We group them into clusters using a distance-based clustering algorithm, such as:
    - DBSCAN (recommended)
    - Simple agglomerative clustering based on pairwise distances

How clustering works

- Two head points belong to the same cluster if they are within R distance from each other.
- Clusters represent groups of people who are physically close.

Why DBSCAN works well? DBSCAN naturally handles:

- unknown number of clusters
- irregular crowd shapes

- variable density
- noise/outliers

Filter small clusters - Any cluster with size < N is ignored (too small to be a meaningful gathering).
Output of spatial grouping

For frame t, we get: Zero, one, or several clusters

Each cluster has: size (number of people) , centroid/location, bounding area (optional)

This gives the “where” and “how many” of potential gatherings.

Step 2 - Temporal Consistency (Tracking clusters across frames)
Spatial clusters change over time. Some appear briefly; others persist. To separate real gatherings from short-lived coincidences, CrowdSense:

- Tracks clusters across frames
- Ensures a cluster exists for at least T consecutive seconds

How cluster tracking works?
For frame t and t+1:

- A cluster in frame t is considered the same cluster in frame t+1 if:
    - its centroid moves only slightly (within a position tolerance),
    - AND its member head points overlap or are close to previous members.

This linking creates a temporal chain of clusters.

Starting the timer - If a cluster first meets the size condition (≥ N), start a timer.
Confirming a gathering - If that cluster persists for ≥ T seconds, CrowdSense declares the cluster a true gathering.

Rejecting a transient cluster

If the cluster dissolves before reaching T, it is:

- discarded, or
- made to restart its timer if it reforms.

Step 3 - Maintaining Gathering Identidy Over Time
Once a gathering is confirmed, it is assigned an ID. The system then tracks:

- size changes (count over time)
- centroid (location movement)
- area/dispersion (how spread out it is)
- density (using convex hull or local density)
- any movement trends

This produces a time series of the gathering’s properties. Multiple gatherings can be tracked at once:

- different zones,
- independent clusters,
- evolving sizes.

Section 5 is essential because:

- It allows CrowdSense to know when and where a crowd begins to form.
- It frames each gathering as an object with temporal context.
- It filters out noise (e.g., people just walking past each other).
- It creates the data structure needed for Section 3.6 risk signals:
    - density,
    - growth rate,
    - compaction,
    - flow stagnation.

Without robust gathering detection:

- risk analysis would be unstable,
- false alarms would rise,
- system couldn't track escalation reliably.

6. Risk Signal Construction

For each gathering (or zone), compute a time series of risk-relevant signals:

1. Density per area
    - People per m² (or relative if no calibration).
    - Rising density, especially beyond ~4–6 people/m², is dangerous.
2. Growth rate
    - Rate of change of count/density (how many people per minute joining).
    - Smoothed over a window to reduce noise.
3. Compaction
    - How area/dispersion changes relative to population.
    - If count ↑ or stable but area ↓ → people packing together.
    - Often measured via convex hull area or average inter-person spacing.
4. Flow stagnation
    - Movement slows or stops in a dense crowd.
    - Approximate via motion of head points or optical flow.
    - High density + low speed over time → congestion risk.
5. Optional signals
    - Bottleneck pressure (high density near exits, choke points).
    - Turbulence (erratic motion/direction changes).

These are kept interpretable: operators can relate them to real conditions ("people are getting stuck", "they're packing tighter", etc.).

7. Predict Phase (Model B: Risk Forecasting)

- Input: Recent window (length W) of risk signal time series for a gathering.
- Output: likelihood of escalation in next X minutes (e.g. 5 minutes).
- Model B options:

1. Rule-based heuristics

    - If density > D_c and flow < V_c for > Y seconds → alert.
    - Or extreme growth rate → warning.
    - Simple, but brittle; used as a baseline.
2. Unsupervised anomaly detection
    - Train on normal (non-incident) crowd data.
    - A model like one-class SVM learns a “safe” envelope.
    - At runtime, if the feature vector is outside the envelope → anomaly → potential risk.
3. Supervised sequence modeling
    - With labeled incident data (or simulations):
        - Use RNN/LSTM/temporal CNN.
        - Input: sequences over W.
        - Output: probability of escalation within X.
    - Powerful but needs incident data; treated as optional when such data is available.

- Training happens offline; deployment is real-time inference, updating predictions every sampled frame.
- Alerts can be explained by referencing which signals are abnormal.

8. Implementation & Configuration

- Hardware: GPU assumed for Model A to maintain near-real-time frame rate; frame rate can be lowered if needed.
- Model A: DM-Count from public code + pretrained weights (e.g., ShanghaiTech, UCF-QNRF).
- Calibration: If known, convert pixels → meters; if not, use pixel-based relative metrics.

Key configurable parameters:

- N – minimum people to count as a gathering.
- R – proximity radius (in meters or pixels).
- T – minimum duration to qualify as a gathering.
- Risk thresholds & Model B sensitivity – trade-off between false alarms and early warnings.

The system runs the loop:
Sense → Detect → Predict each time step, per zone, exposing a dashboard with:

- Crowd counts.
- Active gatherings.
- Risk alerts with their rationale.

[1] Wang, B., Gao, J., Xu, W., & Shen, C. (2020). **Distribution matching for crowd counting**. *Advances in Neural Information Processing Systems (NeurIPS)*. (NeurIPS Proceedings)

[2] Li, Y., Zhang, X., & Chen, D. (2018). **CSRNet: Dilated convolutional neural networks for understanding the highly congested scenes**. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (CVF Open Access)

[3] Helbing, D., Johansson, A., & Zein Al-Abideen, H. (2007). **The dynamics of crowd disasters: An empirical study**. *Physical Review E, 75*(4), 046109. (arXiv)

[4] Bek, S., & Monari, E. (2016). **The crowd congestion level: A new measure for risk assessment in video-based crowd monitoring**. In *2016 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*. (ResearchGate)

[5] Sedky, M., Bamaqa, A., Bosakowski, T., Bastaki, B. B., & Alshammari, N. (2022). **SIMCD: SIMulated crowd data for anomaly detection and prediction**. *Expert Systems with Applications, 203*, 117475. (ScienceDirect)

Appendix:

Paper 1 - CSRNet: Dilated Convolutional Neural Networks for Understanding the Highly Congested Scenes (CVF Open Access) Yuhong Li1,2 , Xiaofan Zhang1 , Deming Chen1

Paper abstract: We propose a network for Congested Scene Recognition called CSRNet to provide a data-driven and deep learning method that can understand highly congested scenes and perform accurate count estimation as well as present high-quality density maps.

The proposed CSRNet is composed of two major components: a convolutional neural network (CNN) as the front-end for 2D feature extraction and a dilated CNN as the back-end, which uses dilated kernels to provide larger receptive fields and replace pooling operations.

CSRNet is an easy-to-train model because of its pure convolutional structure. We demonstrate CSRNet on four datasets (ShanghaiTech dataset, the UCF CC 50 dataset, the WorldEXPO'10 dataset, and the UCSD dataset) and deliver the state-of-the-art performance.

In the ShanghaiTech Part B dataset, CSRNet achieves 47.3% lower Mean Absolute Error (MAE) than the previous state-of-the-art method. We extend the targeted applications for counting other objects, such as the vehicle in the TRANCOS dataset. Results show that CSRNet significantly improves the output quality with 15.4% lower MAE than the previous state-of-the-art approach.

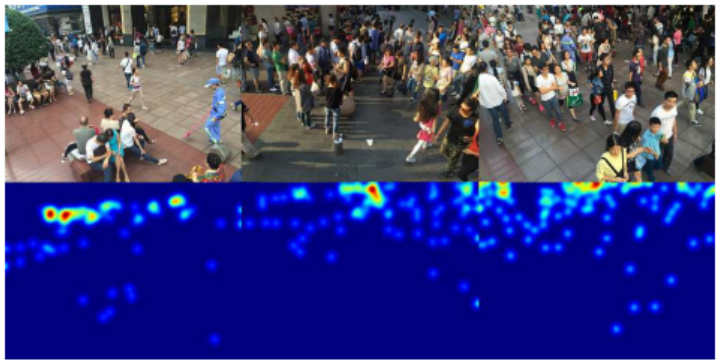


Figure 1. Pictures in first row show three images all containing 95 people in ShanghaiTech Part_B dataset [18], while having totally different spatial distributions. Pictures in second row show their density maps.

Paper 2 - Intelligent Real-Time Crowd Density Estimation for Proactive Event Safety: A Machine Learning Approach, Sheela S Maharajpet, Ananya V Hegde

This paper suggests combining **object detectors with density regressors**. A recent real-time crowd density system for proactive event safety "utilizes YOLOv8 for people detection and CSRNet for density estimation" on live video from CCTV or drones, reporting four discrete density levels (low, medium, high, critical) and a web dashboard for operators. (ResearchGate) These works align closely with CrowdSense's plan to ingest live RTSP CCTV streams and maintain per-zone density estimates in real time.

Paper abstract - High-density crowd events like public concerts, sporting events, or religious festivals represent significant safety challenges due to high crowd density. Methods of monitoring crowds, such as manual observation or passive surveillance, often don't provide real-time information. Therefore, we present a real-time crowd density estimation solution that utilizes YOLOv8 for people detection and CSRNet for density estimation.

The crowd density estimation system uses live video feeds from surveillance cameras or drone footage. The system will assess crowd density at four levels across four distinct areas: Low, Medium, High, and Critical. The density estimator has a web-based dashboard that provides real-time analytics reports, heatmap density estimates, and historical records, which can assist in making quick, informed decisions and assessments following an event.

The system is validated on benchmark datasets and real-world video streams with 95.3% detection accuracy, 7.4 MAE in crowd counting, and 28 FPS processing with off-the-shelf GPU hardware. The results show high accuracy with low latency, making it feasible for real-world applications for large-scale events. The main contributions of the work include using YOLOv8 integrated with CSRNet to jointly detect and estimate crowds, developing a real-time dashboard to provide transparent crowd analytics, and system validation with quantitative metrics and real-world evidence.

The proposed system combines machine learning with computer vision and provides
The capability for real-time crowd density estimation across various zones in an event venue.

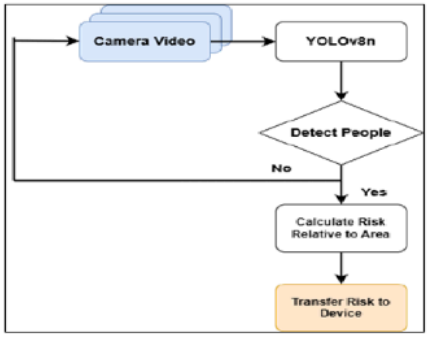


Figure 3. Overall Frame of the Real-time Crowd Density Estimation and Stampede Risk Assessment System

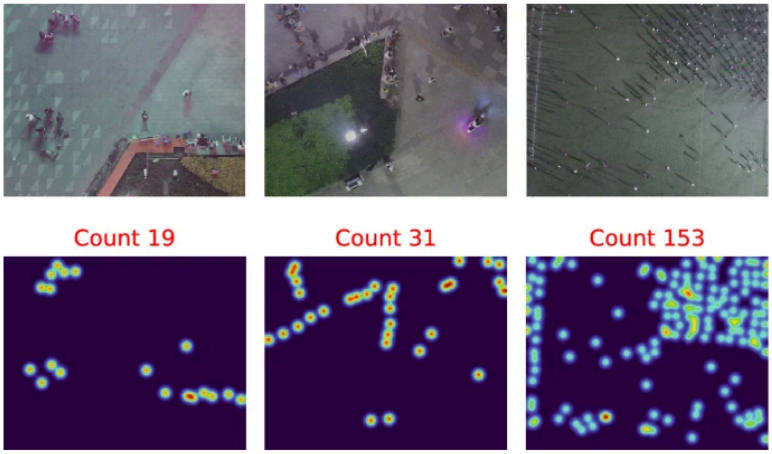
Paper 3 - LCDnet: a lightweight crowd density estimation model for real-time video surveillance. Muhammad Khan, Hamid Menouar, Ridha Hamila.

This work focuses on improving **real-time suitability** and deployment constraints. It introduces LCDnet, “a lightweight crowd density estimation model … for real-time video surveillance,” specifically targeting drone platforms with limited compute and using curriculum learning to stabilise training.

Paper Abstract - Automatic crowd counting using density estimation has gained significant attention in computer vision research. As a result, a large number of crowd-counting and density-estimation models using convolutional neural networks (CNNs) have been published in the last few years. These models have achieved good accuracy over benchmark datasets. However, attempts to improve accuracy often increase the complexity of these models.

In real-time video surveillance applications using drones with limited computing resources, deep models incur intolerably higher inference delay. In this paper, we propose (i) a Lightweight Crowd Density estimation model (LCDnet) for real-time video surveillance, and (ii) an improved training method using curriculum learning (CL). LCDnet is trained using CL and evaluated over two benchmark datasets, i.e., DroneRGBT and CARPK. Results are compared with existing crowd models. Our evaluation shows that LCDnet achieves reasonably good accuracy while significantly reducing inference time and memory requirements, and thus can be deployed on edge devices with minimal computing resources.

Fig. 2



Sample images (top) and their corresponding density maps (bottom) from DroneRGBT dataset

References

[1], [2], [4], [5], [7] - Bipartite Matching for Crowd Counting with Point Supervision Hao Liu1,2,3 , Qiang Zhao1 , Yike Ma1 and Feng Dai1 - Link
[3] - Distribution Matching for Crowd Counting - Boyu Wang∗ Huidong Liu∗ Dimitris Samaras Minh Hoai - Link