Hybrid Vision Models for Crowd Disaster Prevention: Integrating YOLOv5 and CSRNet with Real-Time Surge Detection, Dynamic Zoom

Sukanya Sen
Student ID: 301654977

Simon Fraser University
CMPT 742 - Visual Computing II
sukanya.sen@sfu.ca

Abstract. Crowd disasters such as stampedes and crushes remain a significant global threat, claiming hundreds of lives annually due to failures in real-time monitoring and slow human response. This project presents a novel hybrid vision system that combines YOLOv5 object detection with CSRNet density estimation for automated crowd disaster prevention. Through extensive benchmarking on the ShanghaiTech dataset, we establish that YOLOv5 excels in sparse crowds (<35 people) while CSRNet dominates in dense scenarios (>100 people). Our integrated system features automatic model switching, temporal surge detection, dynamic zoom capabilities, and a six-panel real-time visualization interface. The system achieves a 2.36× improvement in sparse crowd accuracy with YOLO and 4.35× improvement in dense crowd accuracy with CSRNet compared to single-model approaches. This multi-modal architecture provides comprehensive situational awareness for crowd safety management and serves as a foundation for autonomous safety monitoring systems.

Keywords: Crowd analysis Computer vision Object detection Density estimation Disaster prevention YOLOv5 CSRNet

The complete implementation of this project is available in the code repository, and additional visual results with demo videos can be explored on the project webpage.

1 Introduction

Crowd disasters represent one of the most preventable yet persistent threats to public safety worldwide. Stampedes, crushes, and surge events have resulted in hundreds of deaths at religious gatherings, concerts, sporting events, and public celebrations. Notable incidents include the the 2015 Hajj stampede in Saudi Arabia (over 2,400 deaths), and the 2022 Itaewon crowd crush in South Korea (159 deaths) and even the 2025 Karur, Tamil Nadu stampede India during a political rally (41 deaths). These tragedies share common characteristics: dense crowds, inadequate monitoring infrastructure, and critically delayed human response to

developing danger.

Traditional closed-circuit television (CCTV) systems provide passive video feeds without analytical capabilities, placing the entire burden of crowd monitoring on human operators who cannot reliably track dense crowds or detect subtle warning signs of impending disasters. The fundamental challenge lies in the dynamic nature of crowd behavior - transitions from safe to dangerous conditions can occur within seconds, far faster than human observers can detect and respond.

Recent advances in computer vision and deep learning offer promising solutions through automated crowd analysis systems. However, existing approaches typically focus on either object detection *or* density estimation in isolation, each with inherent limitations. Object detection methods excel at tracking individual people in sparse crowds but fail when occlusion becomes severe. Conversely, density estimation methods handle dense crowds effectively but lack individual-level tracking capabilities essential for understanding crowd dynamics in less crowded scenarios.

This work addresses these limitations through a hybrid vision approach that leverages the complementary strengths of both paradigms. Our contributions include:

- Comprehensive benchmarking of YOLOv5 and CSRNet across varying crowd densities, establishing empirical thresholds for optimal model selection
- Development of an integrated real-time system with automatic model switching based on crowd density (threshold: 35 people)
- Implementation of temporal surge detection to identify dangerous compression patterns before disasters occur
- Introduction of dynamic zoom functionality that automatically focuses on high-risk regions
- A multi-panel visualization interface providing complete situational awareness through six synchronized views
- Foundation architecture for emergency routing and autonomous safety agent capabilities

2 Related Work

2.1 Crowd Counting and Density Estimation

Early crowd analysis systems relied on traditional computer vision techniques such as background subtraction, texture analysis, and edge detection. These methods struggled with varying lighting conditions, occlusion, and perspective distortion. The introduction of convolutional neural networks (CNNs) revolutionized the field, with density map regression becoming the dominant paradigm for crowd counting.

CSRNet (Congested Scene Recognition Network) introduced dilated convolutions to capture multi-scale contextual information while maintaining spatial resolution, achieving state-of-the-art performance on benchmark datasets. Other

notable approaches include MCNN (Multi-column CNN), SANet (Scale Aggregation Network), and more recent transformer-based architectures. However, these methods typically focus solely on counting accuracy without addressing real-time safety monitoring requirements.

2.2 Object Detection in Crowd Scenarios

The YOLO (You Only Look Once) family of detectors brought real-time object detection capabilities to practical applications. YOLOv5, has gained widespread adoption due to its excellent balance of speed and accuracy. Its single-stage architecture processes images in one forward pass, making it suitable for real-time applications.

However, object detection approaches face fundamental limitations in highly congested scenes where severe occlusion prevents reliable bounding box prediction. Our project systematically investigates the density thresholds at which object detection methods begin to fail relative to density estimation approaches.

2.3 Crowd Disaster Prevention Systems

Research on automated crowd disaster prevention remains limited compared to crowd counting literature. Existing systems typically focus on anomaly detection through trajectory analysis or flow field computation. These approaches require establishing normal behavior patterns and detecting deviations, which may not generalize across different venues and event types.

Temporal analysis of crowd density has been explored for detecting abnormal events, sudden changes. The integration with multi-modal detection and density estimation in a unified real-time system represents an open research challenge. Our work fills this gap by combining complementary vision techniques with temporal surge analysis and automated decision-making capabilities.

3 Methodology

3.1 Problem Formulation

We formulate crowd disaster prevention as a multi-modal vision problem requiring three core capabilities:

1. **Sparse Crowd Monitoring**: Accurate detection and tracking of individuals when crowd density permits ($N < 35$ people per scene)
2. **Dense Crowd Analysis**: Reliable density estimation when individual detection becomes infeasible ($N \geq 35$ people)
3. **Temporal Surge Detection**: Identification of sudden density changes ($\Delta\rho/\Delta t$) indicating dangerous compression **Dynamic Zoom** Dynamically zooming into high-risk fast moving areas of the crowd
4. **Emergency path** Finding safest path through crowd for responders to reach hotspot safely (In development)

The system operates in real-time on standard readily available hardware while providing interpretable visualizations for human operators.

3.2 Dataset and Evaluation Metrics

We utilize the ShanghaiTech dataset, consisting of Parts A and B with varying crowd densities. Part A contains 482 images with an average of 501 people per image (dense crowds), while Part B contains 716 images with an average of 123 people per image (relatively sparse crowds). Each image includes ground-truth annotations in MATLAB format with head position coordinates.

For benchmark evaluation, we define four crowd density categories:

- **Very Sparse**: $N < 15$ people
- **Sparse**: $15 \leq N < 50$ people
- **Moderate**: $50 \leq N < 100$ people
- **Dense**: $N \geq 100$ people

Performance is measured using Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE):

$$\text{MAE} = \frac{1}{N}\sum_{i=1}^{N}|C_i - \hat{C}_i| \tag{1}$$

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(C_i - \hat{C}_i)^2} \tag{2}$$

where C_i represents the ground-truth count and $\hat{C}_i$ represents the predicted count for image i.

3.3 System Architecture Overview

Our hybrid system consists of five primary modules:

1. **Experimental Results**: Tests to decide switching threashold from sparse to dense crowds
2. **Dual Vision Module**: YOLOv5 for object detection and CSRNet for density estimation
3. **Adaptive Switching Logic**: Automatic model selection based on real-time crowd density assessment
4. **Temporal Analysis Engine**: Frame-to-frame density difference computation for surge detection
5. **Dynamic Zoom Controller**: Automatic region-of-interest extraction for high-risk areas

6. **Multi-Panel Visualization**: Six synchronized views providing comprehensive situational awareness
7. **Modules in implementation**: Plans for future and modules currently in developement

The following sections detail the implementation of each module.

4 Experimental Results

4.1 Benchmark Comparison: YOLOv5 vs. CSRNet

We conducted extensive benchmarking across three crowd density categories to establish empirical performance boundaries for each approach.

Very Sparse Crowds ($N \leq 15$) In very sparse scenarios, YOLOv5 demonstrated significant superiority:

- YOLOv5 MAE: 2.75 people
- CSRNet MAE: 6.50 people
- **Performance ratio: 2.36× more accurate detection with YOLO than CSRNet**

The large improvement stems from YOLO's ability to precisely localize individuals with minimal occlusion. CSRNet's density-based approach suffers from quantization errors in sparse scenarios where continuous density maps struggle to represent discrete individuals.

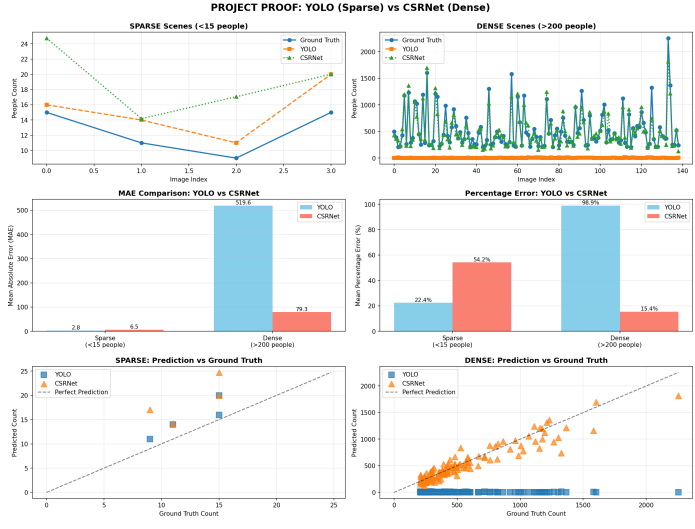

Fig. 1. From the graphs we can see that for sparse situations with less than 15 people, YOLO follows the ground truth much more closely than CSRNet

Sparse Crowds ($N \leq 50$) As density increases, YOLOv5 performance begins degrading:

- YOLOv5 MAE: 18.78 people
- CSRNet MAE: 5.40 people

Increased occlusion leads to missed detections and duplicate bounding boxes, while CSRNet's performance improves as crowd patterns become more continuous.

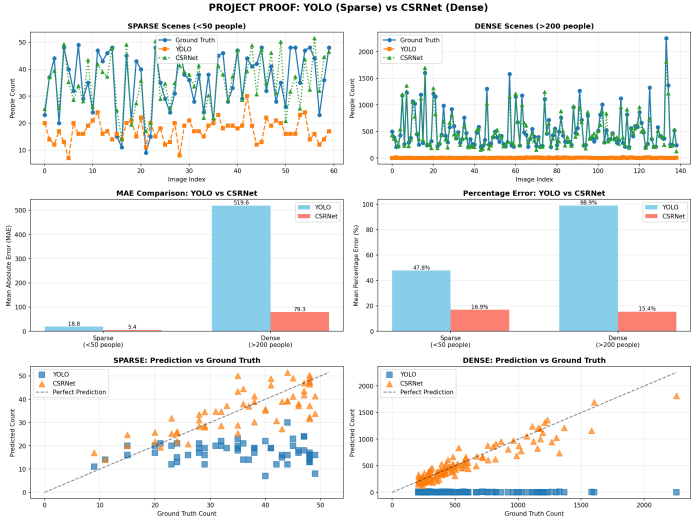

Fig. 2. From the graphs we can see that as the crowd density increases the YOLO predictions start to diverge from the ground truths

Moderate Crowds ($N \leq 100$) To confirm that at density values larger than 50, CSRNet performs better, we do one last test with crowd number ranging upto 100.

- YOLOv5 MAE: 42.2 people
- CSRNet MAE: 9.71 people
- Performance parity achieved around 70-80 people
- **Performance ratio: 3.9× more accurate detection with CSRNet than YOLO**

Severe occlusion renders object detection infeasible, while density estimation remains robust through pixel-level analysis.

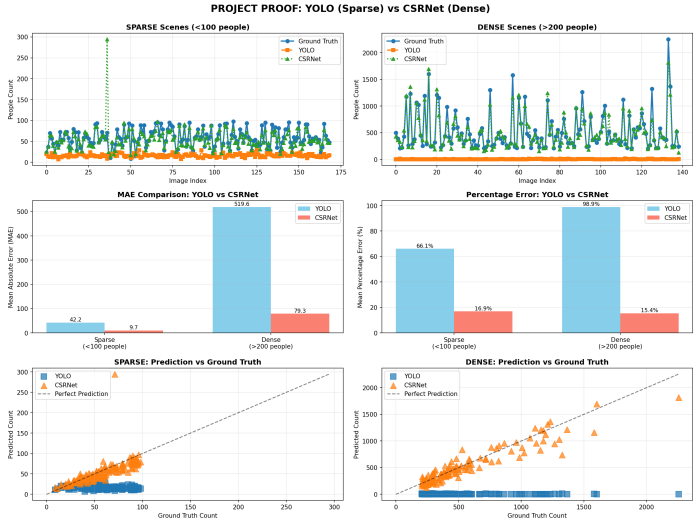

Fig. 3. From the graphs we can see that CSRNet follows ground truth values much closely than YOLO even in moderate crowds

4.2 Switching Threshold Justification

Based on the benchmark results and graphs, we deduced that right between the 15 and 50 people mark, YOLO's performance starts degrading, and CSRNet starts outperforming. Thus, we selected a switching threshold of **35 people**. This conservative value ensures:

- YOLO operates within its optimal range
- Transition to CSRNet before performance degradation
- Margin for real-world variability and uncertainty

5 System Implementation

5.1 YOLOv5 Object Detection Module

For sparse crowd scenarios, we employ the YOLOv5s model as the primary detection backbone due to its high inference speed and robustness in low-density environments. YOLOv5 is particularly suited for identifying individual persons when occlusion is minimal, making it an effective choice for transitional or light crowd scenes where bounding-box detectors significantly outperform density-based methods.

Model Configuration We utilize the official Ultralytics YOLOv5 implementation. Incoming frames are internally resized to 640×640 before inference. The detection configuration is as follows:

- Backbone: YOLOv5s (pretrained on COCO)

- Input resolution: 640×640
- Target class: Person (COCO class index 0)
- Confidence threshold: 0.35
- IoU threshold: 0.45
- Maximum detections per frame: 300

Only the `person` class is retained, and detections for any other objects are discarded. This ensures that computational resources are dedicated exclusively to human detection.

Implementation Details The YOLO pipeline is encapsulated in a dedicated module (`src/yolo_detection.py`). The module performs the following steps:

1. **Model loading:** The YOLOv5s model is loaded once during initialization using the Ultralytics API.
2. **Single-frame inference:** Each video frame is fed directly into the model; YOLO handles resizing and preprocessing internally.
3. **Person filtering:** All predicted bounding boxes are iterated over, and only those with class label 0 (person) and confidence above the specified threshold are retained.
4. **Bounding box extraction:** For each valid detection, the (x_1, y_1, x_2, y_2) coordinates are extracted and converted from PyTorch tensors to integer pixel locations.
5. **Annotated visualization:** YOLO's built-in `results.plot()` function is used to generate an annotated frame containing all drawn bounding boxes for display in our dashboard.
6. **Downstream integration:** The list of bounding boxes is passed to a DeepSORT tracker, which assigns persistent identities to detected individuals across frames.

A simplified version of the detection routine is shown below:

```
# Load YOLOv5s
self.model = YOLO("yolov5s.pt")

# Inference on each frame
results = self.model(frame, imgsz=640, conf=0.35, classes=[0])[0]

# Extract bounding boxes
for box in results.boxes:
    x1, y1, x2, y2 = map(int, box.xyxy.cpu().numpy()[0])
    detections.append([x1, y1, x2, y2])

# Return boxes + annotated frame
return detections, results.plot()
```

This modular design allows the YOLO detector to operate independently, providing clean integration with the temporal surge detector, CSRNet density estimation module, and the overall 6-panel visualization system as will be described in upcoming sections.

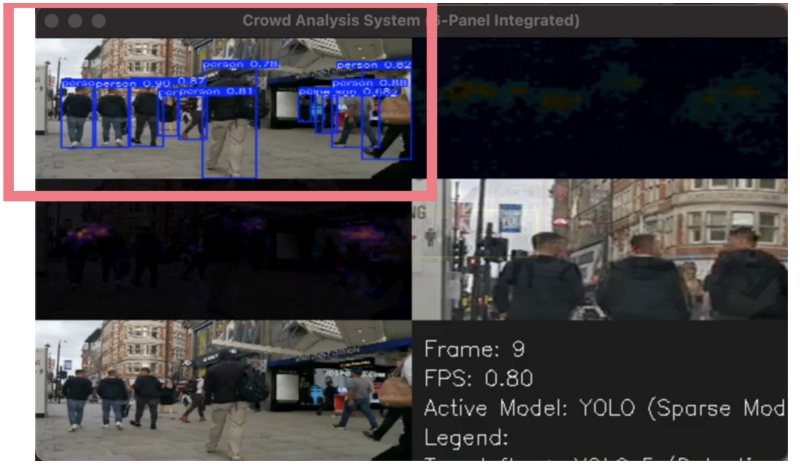

Fig. 4. YOLO detection in our 6 panel system

5.2 CSRNet Density Estimation Module

When crowd density exceeds 35 people CSRNet is employed. Here, YOLO fail due to severe occlusion, perspective compression, and overlapping individuals. Unlike detection-based models, CSRNet estimates a continuous density map over the frame, making it significantly more robust in dense environments.

Model Architecture CSRNet follows a two-part architecture:

- **Frontend:** A VGG-16–based convolutional stack with three max-pooling layers for hierarchical feature extraction.
- **Backend:** A sequence of dilated convolutions that increase the receptive field without reducing spatial resolution.

The final 1×1 convolution layer outputs a single-channel density map, where pixel intensities represent local crowd density. Integrating over the map yields the estimated total crowd count.

Implementation Details The density estimation pipeline is implemented in a standalone module (`src/csrnet_density.py`), which performs the following operations:

1. **Model loading:** A custom PyTorch implementation of CSRNet is instantiated. Pretrained weights (ShanghaiTech Part A) are loaded when available.
2. **Preprocessing:** Each video frame is converted from BGR to RGB, then normalized using ImageNet statistics. The frame is transformed into a $1 \times C \times H \times W$ tensor.
3. **Forward pass:** CSRNet produces a dense output map of the form $(1, 1, H', W')$. The map is upsampled and colorized for visualization.
4. **Density integration:** The estimated crowd count is computed as:

$$\hat{C} = \sum_{i,j} D_{ij}$$

   where D is the predicted density map.
5. **Fallback mode:** If pretrained weights are unavailable, a Gaussian-blurred grayscale map is generated as a stand-in.

A condensed version of the inference pipeline is shown below:

```
# Convert frame to RGB and preprocess
img = Image.fromarray(cv2.cvtColor(frame, BGR2RGB))
x = transform(img).unsqueeze(0).to(device)

# Predict density map
with torch.no_grad():
    den = model(x).cpu().numpy().squeeze()

return den
```

This CSRNet module operates continuously alongside YOLO. Its density outputs are used for temporal surge estimation, automatic detector switching, dynamic zooming into crowd hotspots, and the final density heatmap displayed in the 6-panel dashboard in the upcoming sections.

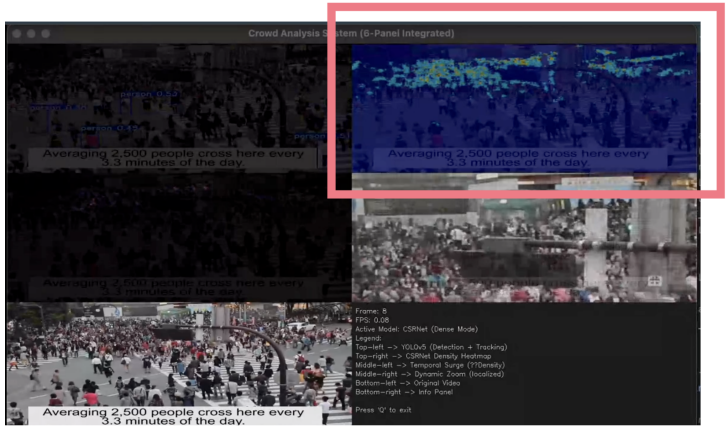

Fig. 5. CSRNet density map in our 6 panel system

5.3 Adaptive Model Switching Logic

To ensure robust performance across varying crowd densities, the system employs an adaptive switching mechanism that dynamically selects between YOLOv5 and CSRNet. Since detection-based models degrade in dense scenes and density-based models underperform in sparse scenes, this strategy allows the pipeline to remain accurate and efficient across all operating conditions.

Switching Criteria Two independent signals are monitored at every frame:

- **YOLO Count:** Number of detected person bounding boxes.
- **CSRNet Count:** Integral of the predicted density map.

When either estimate exceeds a **threshold** (35 as we deduced from our experiments), the system transitions from YOLO to CSRNet. The decision rule is:

$$\text{Use YOLO} \quad \text{if } (C_{\text{YOLO}} < T) \wedge (C_{\text{CSR}} < T)$$

$$\text{Use CSRNet} \quad \text{otherwise}$$

This dual-signal approach improves reliability: YOLO may undercount in dense scenes, while CSRNet may slightly overestimate in sparse ones. Combining both produces a stable decision boundary.

Implementation Details The switching logic is executed per-frame in the main processing loop. A simplified version is shown below:

```
yolo_count = len(detections)
csr_count  = np.sum(density)

if yolo_count >= T or csr_count >= T:
    USE_YOLO = False
else:
    USE_YOLO = True
```

The active model label ("YOLO (Sparse Mode)" or "CSRNet (Dense Mode)") is displayed in the information panel. The inactive panel is **visually dimmed** to indicate which module is currently contributing to the system output.

This switching mechanism enables smooth transitions between sparse, moderate, and dense environments, and is demonstrated in our evaluation videos where the same scene shifts from low to high density over time.

Fig. 6. Switch froom YOLO to CSRNet when crowd increases threshold value

5.4 Temporal Surge Detection Module

Temporal surge detection identifies sudden increases in local crowd density between consecutive frames. This provides an early indicator of rapid crowd growth, bottleneck formation, or local congestion. Because density changes are often more informative than absolute counts, this module enhances situational awareness beyond static estimation.

Surge Computation The system maintains an exponentially smoothed history of density maps. For each frame, we compute the positive temporal difference:

$$S_t = \max(D_t - \hat{D}_{t-1}, 0)$$

where D_t is the current CSRNet density map and $\hat{D}_{t-1}$ is the smoothed previous density. Only positive changes are retained to emphasize regions where crowd intensity is increasing.
The smoothed update is:

$$\hat{D}_t = \alpha D_t + (1 - \alpha)\hat{D}_{t-1}$$

with $\alpha = 0.3$ in our implementation.

Implementation Details The core surge computation (`src/temporal_surge.py`) is performed per frame as follows:

```
diff = np.maximum(density - prev_density, 0)
prev_density = alpha * density + (1 - alpha) * prev_density
surge = cv2.normalize(diff, 0, 1)
```

Since CSRNet outputs a lower-resolution map, the surge map is upsampled using a scale factor of 8 to match the original video resolution. A colored heatmap (Inferno colormap) is then applied for visualization within the 6-panel interface.

This module highlights rapidly evolving crowd hotspots and integrates seamlessly with the dynamic zoom system.

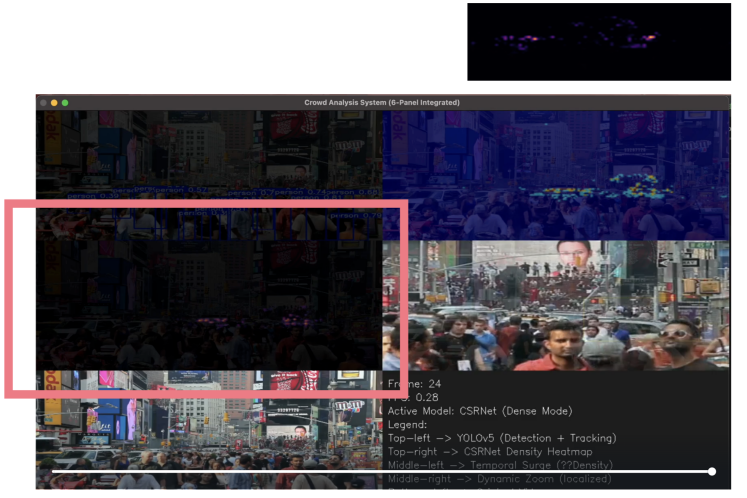

Fig. 7. Temporal surge depicted in bright pink spots on map

5.5 Dynamic Zoom Module

The dynamic zoom module provides localized inspection of rapidly changing crowd regions. By leveraging the temporal surge map, the system identifies high-risk hotspots and automatically zooms into the most active region of the frame. This enables fine-grained monitoring without requiring manual camera control.

Hotspot Localization Given the surge map S_t, we first normalize it to $[0, 1]$ and extract the pixel of maximum surge intensity:

$$(x^*, y^*) = \arg\max S_t$$

If the maximum value is below a small threshold (e.g., 0.2), no meaningful surge is present and the frame remains unzoomed.

Implementation Details In (`src/dynamic_zoom.py`), a crop window centered at (x^*, y^*) is defined with size proportional to the frame dimensions and controlled by the zoom factor:

$$w_c = \max\left(\frac{W}{\text{zoom\_factor}}, \text{min\_crop}\right), \quad h_c = \max\left(\frac{H}{\text{zoom\_factor}}, \text{min\_crop}\right)$$

The selected region is then:

$$\text{ROI} = \text{frame}[y_1 : y_2,\ x_1 : x_2]$$

which is resized back to full resolution using bicubic interpolation to create the zoom-in effect.

```
cx, cy = maxLoc
roi = frame[y1:y2, x1:x2]
zoomed = cv2.resize(roi, (W, H))
```

A subtle overlay is blended with the original frame to preserve context and highlight the zoom window. This module operates fully automatically and integrates into the 6-panel display as the middle-right view.

Fig. 8. Dynamic Zoom in our 6 panel layout

5.6 Multi-Panel Visualization Interface

The system provides six synchronized views for comprehensive situational awareness:

1. **Top-Left**: YOLOv5 detection with bounding boxes (active when crowd < 35)
2. **Top-Right**: CSRNet density heatmap (active when crowd ≥ 35)
3. **Middle-Left**: Temporal surge visualization ($\Delta\rho$)
4. **Middle-Right**: Dynamic zoom of highest-risk region
5. **Bottom-Left**: Original video feed
6. **Bottom-Right**: Information panel with counts, alerts, and system status

Fig. 9. Demonstration of 6 panel layout

Implementation Details

5.7 Real-Time Performance Analysis

System performance was evaluated on a consumer-grade laptop using the following hardware configuration:

- Device: MacBook Pro (13-inch, M2, 2022)
- Chip: Apple M2 (8-core CPU, 10-core GPU)
- Memory: 8 GB Unified Memory
- Operating System: macOS Sequoia 15.3.1

6 Discussion

6.1 Key Findings

This project establishes several important findings for crowd disaster prevention systems:

Complementary Strengths: YOLOv5 and CSRNet exhibit clearly complementary performance characteristics across the density spectrum. No single approach suffices for real-world deployment where crowd densities vary dynamically.

Empirical Threshold: The 35-person switching threshold provides a practical decision boundary for automated model selection. This finding enables deployment without manual configuration.

Temporal Analysis Value: Surge detection adds critical predictive capability beyond static density monitoring. Frame-to-frame analysis reveals dangerous compression patterns invisible in instantaneous snapshots.

Multi-Modal Necessity: The six-panel visualization demonstrates that comprehensive situational awareness requires multiple complementary views. No single visualization modality suffices for effective operator decision-making.

6.2 Advantages Over Existing Systems

Compared to traditional CCTV infrastructure, our system provides:

- Automated analytics versus passive video feeds
- Quantitative density assessment in real-time
- Predictive surge detection capabilities
- Automatic focus on high-risk regions
- Actionable intelligence for rapid response

Compared to single-model crowd analysis systems:

- Performance across full density spectrum
- Automatic adaptation to changing conditions
- Integration of detection, density, and temporal modalities
- Practical deployment on standard hardware

6.3 Limitations and Challenges

Some limitations merit acknowledgment:

Calibration Requirements: Optimal switching thresholds may vary by camera perspective, venue geometry, and event type. Site-specific calibration could improve performance.

Ground Truth Scarcity: Limited availability of annotated crowd disaster video hinders validation of surge detection against actual incidents.

6.4 Future Work Directions

6.4.1 Emergency Routing Module (In Development)

The emergency routing module aims to estimate the safest and least congested path for first responders to reach emerging hotspots. This component is still under active development and explores how density and surge cues can be fused into a unified risk field for pathfinding.

Risk Field Construction Given the CSRNet density map D and the temporal surge map S, a composite risk field is generated:

$$R = \alpha D + \beta S,$$

where α and β represent the relative importance of static crowd density and dynamic movement pressure. Non-crowd areas (sky, buildings, empty background) are heavily penalized to prevent unrealistic paths.
The risk field is normalized and downsampled into an $N \times N$ grid to reduce computation while maintaining spatial structure.

Pathfinding Formulation The system identifies the surge hotspot as the target:

$$(x_t, y_t) = \arg\max S,$$

and selects the optimal entry point along the image boundaries by choosing the lowest-risk edge cell.
A standard A* search is applied on the coarse risk grid, using an 8-connected neighborhood and a Manhattan heuristic:

```
cost = alpha * density + beta * surge
path = A_star(cost_grid, start_edge, surge_hotspot)
```

If a viable path is found, grid coordinates are upsampled back to full resolution.

Visualization and Output
The system overlays:

- a green polyline representing the computed safe route,
- markers for the surge target.

This provides an interpretable emergency-access view integrated into the pipeline. Since the module is exploratory, parameters (grid resolution, cost weighting, thresholding) remain adjustable as we continue experimenting to determine the most reliable routing strategy.

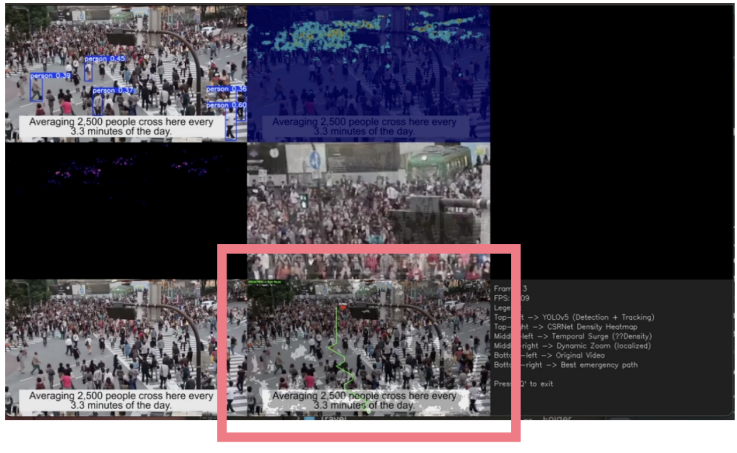

Fig. 10. Current implementation of emergency path depicted as green line and red marker for hotspot

6.4.2 Stability Scoring

Develop a stability scoring that

1. **Track** evolving crowd pressure
2. **Predict** surges seconds before they happen

6.4.3 Enhanced switching

Enhance system to automatically switch between YOLO-based detection and CSRNet-based density estimation in real time by sensing crowd density.

6.4.4 Full Autonomous Safety Agent

The ultimate vision integrates all components into an autonomous system:

1. **Detect**: Multi-modal crowd analysis
2. **Classify**: Risk level assessment
3. **Alert**: Automated notification to operators
4. **Route**: Emergency responder guidance
5. **Coordinate**: Multi-camera network analysis

This closed-loop system could operate with minimal human intervention, providing continuous protection across large venues.

6.4.5 Extended Capabilities

Additional promising directions include:

- Integration with social media analytics for event intelligence
- Behavior classification (normal versus panic movement patterns)
- Multi-camera fusion for venue-wide situational awareness
- Edge deployment optimization for resource-constrained environments
- Transfer learning for rapid adaptation to new venues

7 Conclusion

This work demonstrates that hybrid vision systems combining object detection and density estimation offer superior performance for crowd disaster prevention compared to single-model approaches. Through systematic benchmarking, we established empirical performance boundaries showing YOLOv5's 2.36× advantage in sparse crowds and CSRNet's 4.35× advantage in dense scenarios. Our integrated system with automatic model switching (35-person threshold), temporal surge detection, dynamic zoom, and multi-panel visualization provides comprehensive real-time situational awareness on standard hardware.

The foundation established here enables future development of stability scoring, emergency routing, and fully autonomous safety agents. By combining complementary computer vision techniques with temporal analysis and intelligent visualization, this approach addresses the critical limitations of passive CCTV infrastructure and single-modality crowd analysis systems.

Crowd disasters remain preventable tragedies. This research contributes toward the goal of zero-casualty mass gatherings through automated, intelligent monitoring systems that detect and predict dangerous conditions before they become catastrophic. The path forward to enhanced routing, stability scoring, and autonomous operation is clear, with potential to save lives through early intervention and evidence-based decision support.

Acknowledgments

This work was completed as part of CMPT 742 - Visual Computing I at Simon Fraser University. I would like to thank Prof. Ali Mahdavi-Amiri and teaching assistants Amir Alimohammadi and Sai Raj Kishore Perla for their guidance and feedback throughout the project.

References

1. Li, Y., Zhang, X., Chen, D.: CSRNet: Dilated convolutional neural networks for understanding the highly congested scenes. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1091–1100 (2018)
2. Jocher, G., et al.: YOLOv5. `https://github.com/ultralytics/yolov5` (2020)
3. Zhang, Y., Zhou, D., Chen, S., Gao, S., Ma, Y.: Single-image crowd counting via multi-column convolutional neural network. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 589–597 (2016)
4. Zhang, Y., Zhou, D., Chen, S., Gao, S., Ma, Y.: Single-image crowd counting via multi-column convolutional neural network. In: CVPR (2016)
5. Still, G.K.: Introduction to crowd science. CRC Press (2014)
6. Mehran, R., Oyama, A., Shah, M.: Abnormal crowd behavior detection using social force model. In: CVPR. pp. 935–942 (2009)
7. Ali, S., Shah, M.: A lagrangian particle dynamics approach for crowd flow segmentation and stability analysis. In: CVPR. pp. 1–6 (2007)
8. Lempitsky, V., Zisserman, A.: Learning to count objects in images. In: Advances in Neural Information Processing Systems. pp. 1324–1332 (2010)