Case Report

Ru'ya: A Lightweight AI Model for UAV-Based Crowd Detection and Monitoring

Rabab Alkhalifa *, Nora Aljomuh, Mozoon Alkahlis, Ritaj Alhamli, Sarah Alashgar, Joud Alahmari, Mehwash Farooqui

Posted Date: 27 February 2026

doi: 10.20944/preprints202602.1734.v1

Keywords: deep learning; computer vision; UAVs; crowd monitoring; lightweight models

 doi:10.20944/preprints202602.1734.v1

Case Report

Ru'ya: A Lightweight AI Model for UAV-Based Crowd Detection and Monitoring

Rabab Alkhalifa *, Nora Aljomuh, Mozoon Alkahlis, Ritaj Alhamli, Sarah Alashgar, Joud Alahmari and Mehwash Farooqui

Dept. of Computer Engineering, Imam Abdulrahman bin Faisal University, Dammam, Saudi Arabia
* Correspondence: raalkhalifa@iau.edu.sa

Abstract

The Ru'ya Drone Project focuses on developing an intelligent lightweight UAV-based system capable of performing real-time crowd detection and density estimation using artificial intelligence (AI) and computer vision techniques. The project integrates drone technology with advanced deep learning models such as TensorFlow and PyTorch to enhance public safety, surveillance efficiency, and situational awareness in large gatherings. The system captures aerial footage through onboard cameras, preprocesses the data, and applies AI models to identify and analyze crowd behavior dynamically. A user-friendly interface allows real-time monitoring, while cloud integration ensures efficient data management and scalability. This project contributes to the growing field of AI-driven autonomous systems and aligns with Saudi Vision 2030, promoting smart technologies for safety and urban management. The results demonstrate the potential of combining UAV platforms with AI to achieve accurate, adaptable, and reliable crowd monitoring solutions for real-world applications Github Code: https://lnkd.in/gEt9JZXZ.

Keywords: deep learning; computer vision; UAVs; crowd monitoring; lightweight models

1. Introduction

Large events require managing massive crowds in dynamic and complex environments. Current crowd monitoring methods rely on security guards and fixed CCTV surveillance systems, which can be limited by visibility, slow response times, and human error. These challenges increase safety risks and reduce emergency responses, highlighting and focusing on the need for more advanced and reliable monitoring systems [1]. In response to this need, our project ***Ru'ya***, meaning "*Vision*" in Arabic introduces a multi-environment drone system for crowd management and control. Our project foresight is to build a lightweight autonomous aerial system model capable of operating across different environments to monitor crowds, enhance safety, and deliver real-time data. By combining unmanned aerial vehicle (UAV) technology with artificial intelligence, this system will align with *Saudi Vision 2030*, especially as it will enhance the user experience through rapid real-time response and wide aerial visibility. Also, existing UAV-based detection solutions, while promising, typically require high GPU resources and are still struggling to provide reliable real-time performance in outdoor conditions. These challenges highlight the need for an improved and more adaptive monitoring approach. Our AI-driven UAV solution is designed to avoid and prevent such accidents by detecting and forecasting crowd densities in real time, providing authorities with early warnings to tackle such situations. Our project intention and expectation are to support authorities such as the Saudi Data and Artificial Intelligence Authority (SDAIA) and event organizers in improving safety measures, optimizing resources, and ensuring smoother large-scale event operations, with the aspiration of being part of the kingdom's journey.

 doi:10.20944/preprints202602.1734.v1

2. Related Work

Recent UAV-based crowd detection systems have predominantly relied on heavyweight deep learning architectures that prioritize detection accuracy over computational efficiency. These include object detection frameworks (e.g., YOLO-based models), graph-embedded convolutional networks, and encoder–decoder density estimation architectures.

Tzelepi and Tefas [2] proposed a graph-embedded CNN that combines Softmax and Euclidean losses for UAV-based crowd heatmap generation, achieving nearly 95% accuracy under cross-validation. Herrera et al. [3] employed YOLOv5 for large-scale aerial monitoring using over 35,000 UAV images, reporting 91–95% counting accuracy. Density-based methods have also been explored: Castellano et al. [4] integrated spatial graphs with fully convolutional networks (FCNs) for aerial crowd estimation, while Elharrouss et al. [5] introduced Drone-SCNet, a cascaded density refinement architecture to address scale variation. Zhao et al. [6] further proposed a hybrid point–density network combining localization and counting within a unified framework. Although these approaches demonstrate strong performance, their computational complexity and reliance on high-end GPUs limit their suitability for onboard UAV deployment.

To address hardware constraints, recent studies have focused on lightweight architectures optimized for embedded systems. Papaioannidis et al. [7] developed a multi-task CNN for UAV safety applications, achieving 86% accuracy while remaining compatible with embedded platforms. Khan et al. [8] introduced LCDNet, a parameter-efficient model designed for real-time inference on Jetson devices. Enhancements to YOLO frameworks have also been investigated: Alhawsawi et al. [9] incorporated contextual augmentation into YOLOv8, and Yallamraju and Jana [10] integrated RegionViT components within a multi-scale YOLOv8 pipeline to balance accuracy and speed.

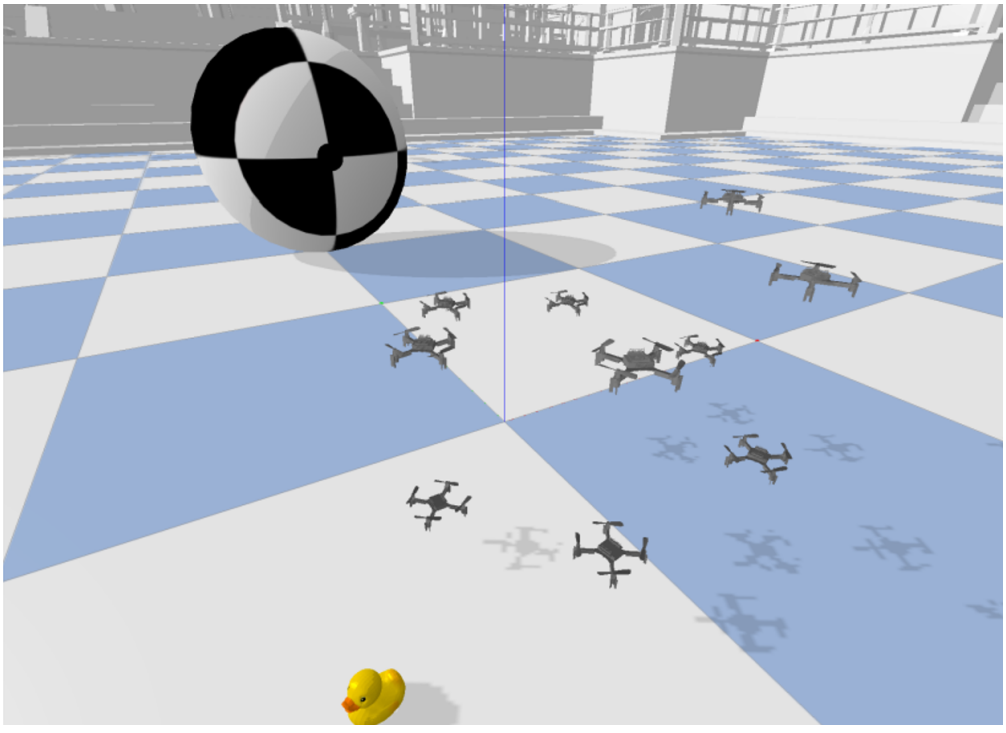

Figure 1. multi-drone simulation in gym-pybullet-drones (reproduced from [11]).

Simulation platforms further support UAV validation. The *gym-pybullet-drones* environment [11] provides realistic quadcopter physics, multi-drone scalability, and OpenAI Gym compatibility, enabling safe and reproducible experimentation (Figure 1). Despite these advances, a gap remains between high-accuracy heavyweight models and deployable lightweight solutions suitable for real-time embedded UAV systems. This work addresses that gap by developing a computationally efficient YOLO-based architecture tailored for practical UAV-based crowd monitoring.

 doi:10.20944/preprints202602.1734.v1

3. Methodology

3.1. System Architecture

Our system follows a layered, modular architecture inspired by the Model–View–Controller (MVC) design pattern to support reliable real-time UAV crowd monitoring, as illustrated in Figure 2.

- **Model (Hardware Subsystem):** The UAV platform and onboard sensors responsible for capturing aerial video and telemetry.
- **Controller (Processing Subsystem):** The AI processing pipeline that pre-processes frames and performs YOLO-based inference to generate detections and crowd analytics.
- **View (Visualization Subsystem):** The operator dashboard that visualizes detections, counts, and density indicators for real-time monitoring.

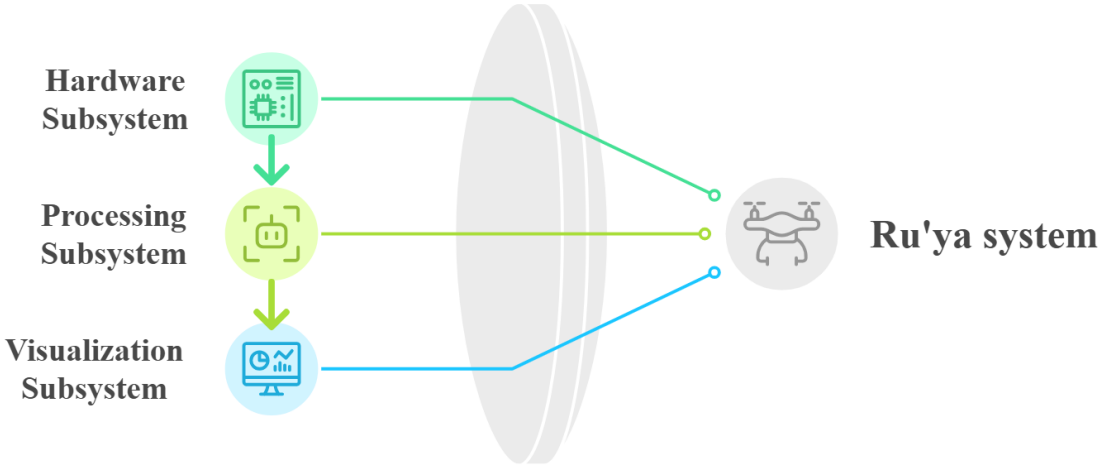

Figure 2. Ru'ya system architecture based on an MVC-inspired modular design.

Each subsystem maintains a clear functional boundary and exchanges data through lightweight communication interfaces, improving maintainability and reducing system coupling. The drone, AI inference engine, and dashboard operate concurrently to enable low-latency crowd detection and analytics from aerial video streams.

3.2. Proposed Lightweight Altitude-Aware Framework

Despite the wide use of deep learning models in crowd detection and monitoring, most studies rarely highlight whether these models are lightweight and suitable for real-time deployment on an edge device such as UAVs. Majority of studies focused on the accuracy of the crowd detection rather than model size, FLOPs, number of parameters, and inference speed. In real deployment, crowd detection and monitoring must operate under limited computing resources and extremely strict time limits. Additionally, through our preliminary experiments we found that the performance of the detection heavily depends on the drone's flight altitude, since the altitude affects the size of the people in the image frame and the quality of the detail of the scene. Accordingly, we found it necessary to evaluate the models on datasets that differ in altitude level and analyze how both crowd detection and computational cost change across these different conditions. Hence, in this paper we studied cross-altitude generalization by training and evaluating the chosen models on datasets with distinct altitude level. By jointly examining the trade-off between accuracy and efficiency, and the altitude-dependent crowd detection behaviour, our work provides real-world UAV deployment evaluation that shows how different models should be chosen for different flight altitudes in order to achieve a good balance between detection accuracy and computational efficiency.

 doi:10.20944/preprints202602.1734.v1

3.3. UAV Crowd Detection Pipeline

The overall processing pipeline is centered on the YOLO (You Only Look Once) deep learning architecture and proceeds as follows:

1. ***Data Acquisition:*** The UAV continuously captures aerial video using a high-resolution onboard camera. Example raw frames illustrating the system input are shown in Figure 3.

Figure 3. Example raw aerial crowd frames used as input to the Ru'ya pipeline.

2. ***Preprocessing:*** Incoming frames are resized and normalized using OpenCV to match YOLO input specifications. Lightweight enhancement techniques, such as Gaussian blurring and color correction, improve robustness under varying illumination and crowd density conditions.
3. ***AI Inference (YOLO-based):*** The YOLO model performs real-time person detection for each frame, producing bounding boxes and confidence scores. Crowd density is estimated by aggregating detections over a grid-based spatial representation.
4. ***Postprocessing and Data Handling:*** Detection outputs (bounding boxes, counts, and density metrics) are formatted into structured JSON objects and transmitted to the backend for aggregation. Results are logged to a database to support both real-time monitoring and historical analysis.
5. ***Visualization and Alerts:*** The dashboard visualizes heatmaps, charts, and real-time statistics using HTML/CSS and JavaScript. Threshold-based alerts are triggered when crowd size or density exceeds predefined safety limits, supporting rapid operational decision-making.
6. ***Concurrency and State Management:*** A multithreaded architecture enables video capture, inference, and dashboard updates to run in parallel. Mutex locks protect shared buffers and prevent race conditions. System states transition as follows: *Idle* $\rightarrow$ *Capturing* $\rightarrow$ *Processing* $\rightarrow$ *Alerting* $\rightarrow$ *Logging*.
7. ***Initialization and Cleanup:*** At startup, YOLO model weights are loaded, compute resources are allocated, and network connections are verified. Upon termination, buffers are flushed, logs are saved, and GPU memory is released to ensure graceful shutdown.

 doi:10.20944/preprints202602.1734.v1

Table 1. Detection performance and inference efficiency of YOLOv8 variants across datasets.

Model	mAP@50	Precision	Recall	Latency (ms)	FPS
DroneCrowd					
YOLOv8s	63.0	70.0	57.0	12.4	80.6
YOLOv8m	32.5	48.4	31.9	29.1	34.3
VisDrone					
YOLOv8s	41.8	54.7	39.4	12.6	79.4
YOLOv8m	41.8	54.7	39.49	29.3	34.1
WiderPerson					
YOLOv8s	59.0	65.0	55.0	12.3	81.3
YOLOv8m	59.3	65.0	55.4	28.9	34.6

4. Experiments and Evaluation

We conducted controlled experiments to evaluate the detection performance, computational efficiency, and deployment suitability of the proposed UAV-based crowd monitoring system. The evaluation focuses on both accuracy metrics and runtime feasibility under consistent training conditions.

1. ***Privacy and Anonymization***
   UAV-based crowd monitoring raises privacy and ethical concerns, particularly in public environments. The proposed system is designed strictly for crowd density estimation and does not perform facial recognition, biometric identification, or individual tracking.
   As a proof-of-concept safeguard, a lightweight anonymization step was implemented during prototyping. Video frames were processed using a pre-trained Haar Cascade face detector, and detected faces were blurred using a large Gaussian kernel to reduce identifiable details (Figure 4). Only anonymized frames were retained for inspection; raw frames were not stored.

Figure 4. Example of face anonymization applied during system prototyping.

This approach is preliminary and not intended as a finalized deployment solution, as face detection in aerial imagery can be unreliable under high altitude, occlusion, and varying viewpoints. The framework follows privacy-by-design principles, emphasizing data minimization and edge-based processing. Any future real-world deployment would require formal ethical approval and

 doi:10.20944/preprints202602.1734.v1

more robust anonymization techniques aligned with applicable regulations, including the Saudi Personal Data Protection Law (PDPL).

2. *Models and Datasets*

We evaluate the detection performance and deployment suitability of YOLOv8-based object detection models for UAV-based crowd monitoring across datasets representing different altitude levels and scene complexities.

Model Variants: Two configurations from the YOLOv8 family are considered, implemented using the official Ultralytics library[1]:

- **YOLOv8s**: a lightweight variant optimized for real-time inference and edge deployment.
- **YOLOv8m**: a medium-sized variant with increased model capacity, designed to improve accuracy at higher computational cost.

Datasets: Evaluation was conducted on three publicly available datasets covering varying UAV altitudes and crowd densities:

- **DroneCrowd**[2]: *high-altitude* UAV crowd imagery.
- **VisDrone**[3]: *medium-altitude* UAV imagery with diverse scenes.
- **WiderPerson**[4]: *low-altitude* person detection with challenging occlusion.

Table 2 lists the dataset sizes and splits used to evaluate cross-altitude behavior.

Table 2. Dataset Statistics and Splits

Dataset	Total	Train	Val	Test
DroneCrowd (Subset)	3000	2100	600	300
VisDrone	10209	6471	548	1610
WiderPerson	13382	8000	1000	4382

Given the large size of the full 33,600-frame DroneCrowd dataset and its computational demands, a randomly sampled subset of 3,000 images was used while preserving scene and crowd-density diversity.

Annotation Scope and Conversion. The three datasets differ in their original annotation formats and label taxonomies; therefore, a unified annotation representation was applied for consistent YOLO training and evaluation.

DroneCrowd annotations were originally provided as head-point coordinates. To enable detector training, each head point was converted into a fixed-size bounding box of 15×15 pixels centered at the annotated location. The box dimensions were selected to approximate the typical person extent at the dataset's scale and were kept constant across all images (not scale-adaptive).

VisDrone and *WiderPerson* provide bounding-box annotations with multiple categories. For alignment with the crowd detection objective, only person-related labels were retained. In VisDrone, the *pedestrian* (class ID 1) and *person* (class ID 2) categories were preserved, while other object classes and ignored regions were excluded. For WiderPerson, pedestrian-related classes (1–3) were retained, while ignore regions (class 4) and crowd labels (class 5) were removed. Ambiguous annotations were discarded to ensure consistent training supervision.

Table 3 summarizes the per-dataset training schedule (epochs, batch size, input size, and precision mode) used for fair comparison. All retained annotations were converted to the YOLO format, i.e., normalized coordinates relative to image width and height. Image resizing was not performed during conversion and was handled during YOLOv8 training (input size 640×640).

[1] https://github.com/ultralytics/ultralytics
[2] https://github.com/VisDrone/DroneCrowd
[3] https://github.com/VisDrone/VisDrone-Dataset
[4] http://www.cbsr.ia.ac.cn/users/sfzhang/WiderPerson/

 doi:10.20944/preprints202602.1734.v1

Table 3. Training Configuration per Dataset.

Dataset	Epochs	Batch Size	Image Size	Precision
DroneCrowd	15	8	640×640	FP32
VisDrone	15	4	640×640	FP32
WiderPerson	30	8	640×640	FP32

Table 4. Hardware Configuration.

Component	Specification
GPU	NVIDIA Tesla T4
Framework	Ultralytics YOLOv8
Pretrained Weights	COCO
Optimizer	SGD
Initial Learning Rate	0.01

3. *Performance Evaluation*

Efficiency Reporting. Since the proposed system targets UAV-based deployment, inference efficiency was evaluated alongside detection accuracy. We report two deployment-oriented metrics: inference latency (milliseconds per image) and throughput (frames per second, FPS).
Latency measures the average time required by the model to process a single input image during inference. It is reported in milliseconds per frame (ms/frame) and reflects the responsiveness of the system in real-time scenarios.
Throughput (FPS) indicates how many frames can be processed per second and is computed as:

$$\text{FPS} = \frac{1000}{\text{Latency (ms)}}.$$

Inference speed was measured under fixed hardware conditions and using a consistent input resolution for all datasets. Therefore, efficiency metrics depend only on the model architecture (YOLOv8s vs. YOLOv8m) and not on dataset content. All measurements were obtained under identical experimental settings to ensure fair comparison between model variants. Latency and FPS were measured based on model inference time only, excluding data loading and preprocessing overhead.
Evaluation Metrics. All experiments were conducted under the fixed hardware and training configuration summarized in Table 4. We evaluate detection performance using mean Average Precision at an Intersection-over-Union (IoU) threshold of 0.5 (mAP@50), together with precision and recall.
A predicted bounding box is counted as a *True Positive (TP)* if it matches a ground-truth box of the same class with IoU ≥ 0.5. Predictions that do not match any ground-truth box are *False Positives (FP)*, and ground-truth boxes that are not matched by any prediction are *False Negatives (FN)*.
The IoU between a predicted box B_p and a ground-truth box B_g is defined as:

$$\text{IoU}(B_p, B_g) = \frac{|B_p \cap B_g|}{|B_p \cup B_g|},$$

where $|B_p \cap B_g|$ is the overlap area and $|B_p \cup B_g|$ is the union area.
For each class c, the Average Precision AP_c is computed as the area under the precision–recall curve obtained by sweeping the detection confidence threshold. The reported mAP@50 is:

$$\text{mAP@50} = \frac{1}{C}\sum_{c=1}^{C} AP_c(\text{IoU} = 0.5),$$

where C is the number of evaluated classes.

 doi:10.20944/preprints202602.1734.v1

Precision and recall are computed as:

$$\text{Precision} = \frac{TP}{TP + FP}, \qquad \text{Recall} = \frac{TP}{TP + FN}.$$

Precision reflects how many predicted detections are correct, while recall reflects how many ground-truth objects are successfully detected.

4. *Simulation Experiments*

Simulation was used to enable controlled testing under repeatable conditions and to analyze the effect of camera height, crowd density, and movement patterns before real-world deployment.

(a) **2D Simulation Environment**

A lightweight 2D simulator was implemented in Python using NumPy for crowd motion modeling and Matplotlib for visualization. Each person was represented as a moving point on a 2D plane. A rectangular region defined the UAV field of view (FoV), and the number of agents inside this region was counted per frame (Figure 5). This setup allowed rapid validation of counting logic and FoV behavior.

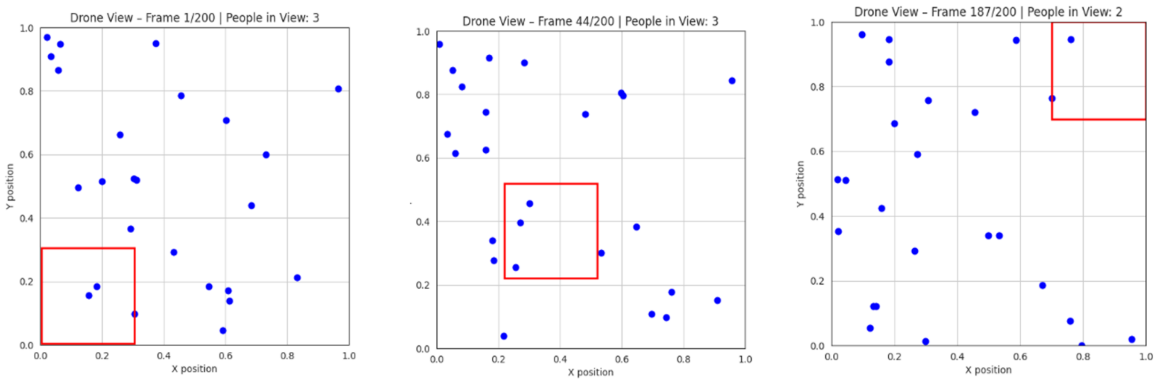

Figure 5. Drone camera field of view in the 2D simulated environment.

(b) **3D Simulation Environment**

To approximate realistic UAV conditions, a 3D environment based on `gym-pybullet-drones` and the PyBullet physics engine was used. Synthetic drone videos were generated and processed through the same YOLOv8 detection pipeline as the real datasets. This ensured consistency between simulation and experimental evaluation while enabling safe preliminary validation (Figure 6).

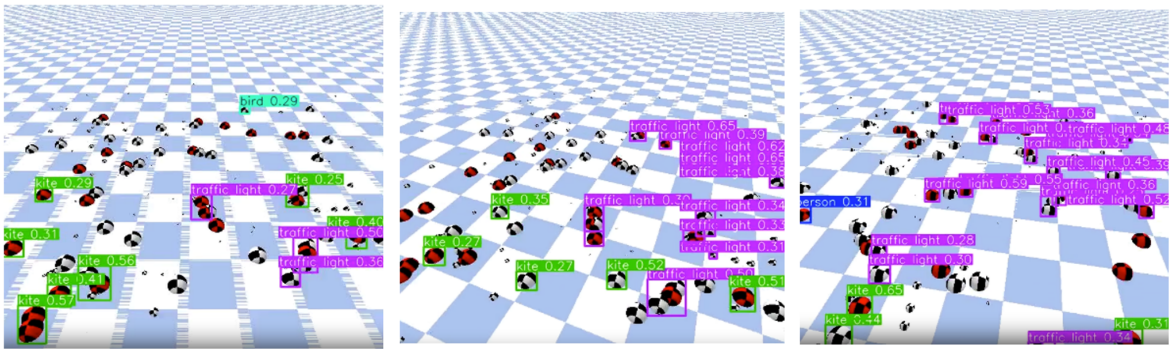

Figure 6. Example YOLOv8 detections on video generated from the 3D simulation environment.

 doi:10.20944/preprints202602.1734.v1

5. Results and Discussion

Table 1 reports accuracy (mAP@50/precision/recall) and efficiency (latency/FPS) for both YOLO variants under identical settings. The results directly support the central objective of this proof-of-concept study: evaluating whether lightweight architectures can provide a better accuracy–efficiency trade-off for UAV-oriented crowd monitoring.

- **Accuracy–Efficiency Trade-off.** Across all datasets, YOLOv8s consistently achieves comparable or superior mAP@50 relative to YOLOv8m, while maintaining substantially lower inference latency. On DroneCrowd (high-altitude imagery), YOLOv8s achieves nearly double the mAP@50 of YOLOv8m, while reducing latency by more than 50%. On VisDrone and WiderPerson, both models exhibit similar detection accuracy; however, YOLOv8s maintains a throughput exceeding 79 FPS compared to approximately 34 FPS for YOLOv8m.
  These findings indicate that *increasing model capacity does not necessarily improve performance in UAV-based crowd monitoring*, particularly when objects appear at small scales.
- **Cross-Altitude Generalization.** The experimental design explicitly evaluates cross-altitude behavior, as introduced in Section 3.2. The superior performance of YOLOv8s on DroneCrowd suggests improved generalization under high-altitude conditions, where individuals occupy minimal pixel regions. Deeper architectures may over-compress spatial features in later layers, reducing sensitivity to small-scale human instances.
  In medium- and low-altitude scenarios (VisDrone and WiderPerson), performance between the two variants converges, *indicating that increased depth does not provide meaningful benefits when object scale is moderately preserved*.
- **Deployment-Oriented Evaluation.** Unlike many prior UAV crowd detection studies that report accuracy alone, this work evaluates latency and FPS under identical hardware conditions. The lightweight YOLOv8s model achieves real-time processing under the tested configuration, indicating stronger suitability for edge-oriented deployment. While onboard UAV validation remains future work, *the results provide reproducible evidence supporting lightweight model selection*.
- **Dataset Suitability Analysis.** The datasets play complementary roles for UAV crowd monitoring. DroneCrowd targets dense *high-altitude* scenes and is most informative for small-object behavior. VisDrone provides *medium-altitude* UAV imagery with diverse scenes, while WiderPerson supports *low-altitude* person detection under challenging occlusion. For consistent benchmarking, all datasets were converted to a unified YOLO format: DroneCrowd head-point annotations were mapped to fixed 15×15 pixel boxes, and only person-related labels were retained for VisDrone and WiderPerson. *This unified conversion enables fair cross-dataset comparison, while emphasizing that altitude and annotation differences must be considered when interpreting results*.
- **Implications for Researchers.** Under the tested configuration, results suggest that lightweight models can offer a stronger accuracy–efficiency trade-off than larger variants in UAV-oriented crowd monitoring, especially in high-altitude small-object scenarios. Therefore, altitude-aware evaluation and efficiency reporting (latency/FPS) may be more informative than parameter scaling alone when selecting UAV detectors.

This reinforces the core contribution of ***Ru'ya***: *an altitude-aware lightweight benchmarking framework that bridges the gap between academic accuracy benchmarks and practical UAV deployment considerations within a proof-of-concept scope*.

6. Conclusions

This paper presented ***Ru'ya***, a framework prototype and feasibility study for UAV-oriented crowd detection using lightweight YOLOv8 models. We benchmarked YOLOv8s and YOLOv8m across three public datasets representing different altitude conditions, and reported both detection metrics and deployment-oriented efficiency (latency and FPS) under consistent settings. The results show that YOLOv8s provides a stronger accuracy–efficiency trade-off, particularly in high-altitude small-object scenarios. We also demonstrated simulation-based validation (2D and 3D) to support controlled

 doi:10.20944/preprints202602.1734.v1

testing of the end-to-end pipeline. Future work will focus on onboard embedded evaluation, improved anonymization suitable for aerial imagery, and real-world flight trials under outdoor operational conditions.

Acknowledgments: The Authors would like to express sincere gratitude to the College of Computer Science and Information Technology at Imam Abdulrahman Bin Faisal University for providing the necessary resources, technical support, and a collaborative environment that facilitated our research and implementation.

References

1. Ludlow Engineers and Associates. Evaluating Drones vs Traditional Techniques in Land Surveying, 2025. Accessed: 2025-09-21.
2. Tzelepi, M.; Tefas, A. Graph Embedded Convolutional Neural Networks in Human Crowd Detection for Drone Flight Safety. *IEEE Transactions on Emerging Topics in Computational Intelligence* **2019**, *5*, 191–204. https://doi.org/10.1109/TETCI.2019.2897815.
3. Herrera, C.; et al. Drone Insights: Unveiling Beach Usage through AI-Powered People Counting. *Drones* **2024**, *8*, 579. https://doi.org/10.3390/drones8100579.
4. Castellano, G.; Castiello, C.; Mencar, C.; Vessio, G. Crowd Detection in Aerial Images Using Spatial Graphs and Fully-Convolutional Neural Networks. *IEEE Access* **2020**, *8*, 64534–64544. https://doi.org/10.1109/ACCESS.2020.2984768.
5. Elharrouss, O.; et al. Drone-SCNet: Sealed Cascade Network for Crowd Counting on Drone Images. *IEEE Transactions on Aerospace and Electronic Systems* **2021**, *57*, 3988–4001. https://doi.org/10.1109/TAES.2021.3087821.
6. Zhao, L.; Bao, Z.; Xie, Z.; Huang, G.; Rehman, Z.U. A Point and Density Map Hybrid Network for Crowd Counting and Localization Based on Unmanned Aerial Vehicles. *Connection Science* **2022**, *34*, 2481–2499.
7. Papaioannidis, C.; Mademlis, I.; Pitas, I. Autonomous UAV Safety by Visual Human Crowd Detection Using Multi-Task Deep Neural Networks. In Proceedings of the Proceedings of IEEE International Conference on Robotics and Automation (ICRA), 2021. https://doi.org/10.1109/ICRA48506.2021.956083.
8. Khan, M.A.; Menouar, H.; Hamila, R. LCDnet: A Lightweight Crowd Density Estimation Model for Real-Time Video Surveillance. *Journal of Real-Time Image Processing* **2023**, *20*. https://doi.org/10.1007/s11554-023-01286-8.
9. Alhawsawi, A.N.; Khan, S.D.; Rehman, F.U. Enhanced YOLOv8-Based Model with Context Enrichment Module for Crowd Counting in Complex Drone Imagery. *Remote Sensing* **2024**, *16*, 4175. https://doi.org/10.3390/rs16224175.
10. Yallamraju, V.R.; Jana, S. Dynamic Object Detection and Tracking System on Unmanned Aerial Vehicles for Surveillance Applications Using RegionViT-Based Adaptive Multi-Scale YOLOv8. *Computational Intelligence* **2025**, *41*. https://doi.org/10.1111/coin.70101.
11. Panerati, J.; Zheng, H.; Zhou, S.; Xu, J.; Prorok, A.; Schoellig, A.P. Learning to Fly—A Gym Environment with PyBullet Physics for Reinforcement Learning of Multi-Agent Quadcopter Control. In Proceedings of the 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2021. https://doi.org/10.1109/IROS51168.2021.9635857.