CROWD DENSITY PREDICTION USING DEEP LEARNING

Prof. Dhananjaya Kumar K[1], Abdul Jabbar Shaikh[2], Ankush K [3], Fathima Zehra[4], Tasmiya[5]

[1] Assistant Professor, Computer science and Engineering, Vidya Vikas Institute of Engineering & Technology, Karnataka, India
[2] Student, Computer science and Engineering, Vidya Vikas Institute of Engineering & Technology, Karnataka, India
[3] Student, Computer science and Engineering, Vidya Vikas Institute of Engineering & Technology, Karnataka, India
[4] Student, Computer science and Engineering, Vidya Vikas Institute of Engineering & Technology, Karnataka, India
[5] Student, Computer science and Engineering, Vidya Vikas Institute of Engineering & Technology, Karnataka, India

ABSTRACT

As urban areas grow and public events become more common, the need for efficient crowd management is increasingly critical—especially at high-traffic locations such as tourist destinations and religious sites. Traditional methods like manual headcounts or static surveillance systems often lack the precision and scalability needed for modern crowd control. To address these limitations, this project introduces a real-time human density monitoring system powered by the YOLOv5 deep learning model. By analyzing video feeds, the system can detect and count individuals, issuing instant alerts via Telegram when crowd density exceeds safe levels. This allows authorities to take prompt action and prevent overcrowding.

The system is implemented using Python and integrates a pre-trained YOLOv5 model within a real-time processing pipeline. Surveillance footage is captured and processed frame-by-frame to detect humans using bounding boxes. The number of detected individuals is used to compute crowd density, and alerts are triggered if it crosses the predefined safety limit. Built in the PyCharm environment, the solution minimizes human involvement while ensuring real-time functionality and high scalability. The alert mechanism utilizes the python-telegram-bot library for seamless communication with relevant personnel.

Keyword: - *Human Detection, Crowd Monitoring, Deep Learning, YOLOv5, Real-Time Analytics, Public Safety, Computer Vision, Surveillance, Telegram Alerts, Density Estimation*

---

1. INTRODUCTION

Managing large crowds in urban areas and during public gatherings has always posed challenges for planners, law enforcement, and event organizers. With increasing urbanization and frequent mass events—such as festivals or visits to popular attractions—the risk of overcrowding has significantly risen. Recent advancements in computer vision and artificial intelligence have led to the development of smarter, automated crowd management tools.

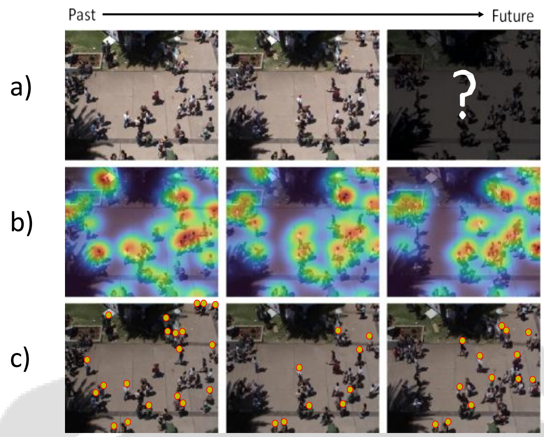

Fig 1: Crowd Density Prediction: Instead of tracking and predicting the movement of each individual person in a crowd, we aim to forecast the overall crowd dynamics. Given (a) a sequence of video frames showing crowd movement, our goal is to predict future scenes in the form of (b) crowd density maps—visual representations indicating how densely populated different areas will be—rather than (c) identifying and forecasting the exact paths of individuals (as represented by the yellow circles).

In particular, object detection models like YOLO (You Only Look Once) have gained traction for their speed and efficiency in real-time environments. YOLOv5, a refined version of this model, offers exceptional accuracy in detecting human figures within video streams. Utilizing YOLOv5 enables automated detection and counting of people in a specific area, helping officials take quick action when crowd thresholds are breached.

2. Literature Review

Crowd density estimation and human detection have seen significant advancements in recent years due to developments in deep learning and computer vision. Traditional methods such as manual counting or infrared sensors provided limited accuracy and lacked scalability. Erden et al. [2] developed a system combining an infrared sensor with a camera to count people, but it was not suitable for real-time applications or dynamic environments. Convolutional Neural Network (CNN)-based approaches have shown promising results in dense crowd counting. Gao et al. [1] proposed SCAR, a spatial-/channel-wise attention regression network, which enhanced the accuracy of density map generation. Liu et al. [7] introduced a deep recurrent spatial-aware network that leveraged temporal features, improving counting accuracy in sequences of frames. Similarly, Marsden et al. [5] implemented fully convolutional networks for highly congested scenes, showing improved robustness in occluded environments.

Regression-based models such as CSRNet focus on estimating a density map directly from input images rather than detecting individual people. While these models provide accurate crowd estimation, they are computationally expensive and often unsuitable for real-time implementation. Yin et al. [6] proposed image processing techniques for crowd estimation using reference images, but such static approaches are not adaptive to dynamic scenes.

To improve processing speed, object detection models like YOLO (You Only Look Once) have gained popularity. YOLO models, particularly YOLOv5, are known for their real-time performance and high accuracy. The architecture offers fast object detection with minimal latency, which is ideal for live surveillance systems. Girshick et al. [4] laid the groundwork for object detection using deep feature hierarchies, which evolved into the YOLO family of models. YOLOv5 builds upon this foundation with optimizations for speed, precision, and deployment efficiency.

Despite these advancements, many systems still lack integrated alert mechanisms and real-time scalability. Most research focuses either on detection or density estimation but not on automatic notification systems. The integration of real-time detection with alert systems, such as Telegram bots, remains limited. This project aims to bridge this gap by using YOLOv5 for real-time crowd detection and integrating Telegram alerts to provide immediate response capabilities during overcrowding situations.

2.1 Aim &Objective

Aim:

To develop a real-time human density detection system using the YOLOv5 deep learning model to enhance crowd management and public safety**.**

Objective:

- To detect and count people accurately in real-time using video feeds.
- To calculate crowd density and identify overcrowding situations.
- To send instant alerts to authorities through a Telegram bot when thresholds are exceeded.
- To minimize human intervention and improve response time during critical crowding events.
- To create a scalable and efficient system suitable for public spaces like religious sites, tourist spots, and transportation hubs.

2.2 Limitations

1. Dependence on Good Lighting Conditions: The system's performance may degrade in low-light or nighttime conditions, as YOLOv5 requires clear visual input to effectively detect humans.High Computational Requirements: Real-time processing with YOLOv5 requires powerful hardware, particularly a good GPU. Systems with lower specifications may experience lag or reduced performance.

2. High Computational Requirements: Real-time processing with YOLOv5 demands high-performance hardware, particularly a capable GPU.

3. Possibility of False Positives and False Negatives: In dense crowds, the system may mistakenly identify non-human objects as people (false positives) or fail to detect some individuals (false negatives), especially in areas with occlusion or high crowd density.

3.Methodology

The entire system is designed to support real-time operations using efficient algorithms and asynchronous messaging. It can be scaled horizontally by running multiple detection instances across different camera feeds.

3.1 System Components:

The system is built using several key components that work together to enable real-time human detection and crowd monitoring. At its core is the YOLOv5 model, chosen for its high accuracy and fast processing capabilities, making it ideal for detecting individuals in dynamic environments. Python serves as the primary programming language, with development and testing carried out in the PyCharm integrated development environment. For communication and alert functionality, the system incorporates the Telegram Bot API, which automatically sends notifications when predefined crowd density thresholds are exceeded. The system processes input from various video sources, including both live surveillance camera feeds and pre-recorded video files, allowing flexibility in deployment and testing.

3.2 Steps Followed

Step 1: Model Selection and Setup: The project began with selecting the YOLOv5 model due to its high accuracy and real-time processing capabilities. The pre-trained YOLOv5 weights were integrated into the system, and necessary Python libraries such as OpenCV, Torch, and python-telegram-bot were installed to ensure the proper functioning of the system.

Step2: Data Preprocessing: Video frames were captured from either live video feeds or pre- recorded videos. These frames were resized and normalized to conform to the input format required by the YOLOv5 model.

Step3: Real-Time Object Detection **:** Each pre-processed frame was passed through the YOLOv5 model for real-time human detection. The model identified individuals within the frame, drawing bounding boxes around each detected person.

Step4: Crowd Density Calculation: The system calculated the crowd density by dividing the number of detected individuals by the area covered by the camera's field of view. A predefined density threshold was established based on safety standards specific to the monitored environment, helping to determine if the crowd density posed a risk.

Step5: Alert Generation: Once the crowd density exceeded the predefined threshold, the system automatically triggered an alert. This alert was sent via a Telegram bot to the designated authorities.

Diagram Representation:

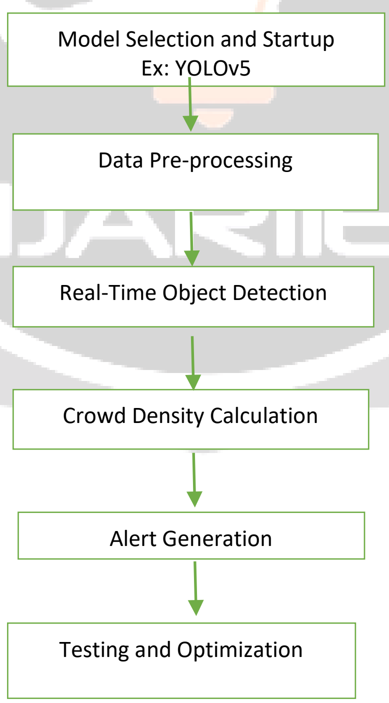

Fig 2: Methodology

4.Implementation & Flow chart

The system is engineered to enable real-time crowd monitoring through the use of optimized algorithms and asynchronous communication mechanisms. It supports scalability by allowing multiple detection processes to operate concurrently across various camera feeds, facilitating broader coverage.

4.1 Development Workflow:

The implementation process begins with the installation of essential libraries such as PyTorch, OpenCV, and Telegram bot APIs. The development environment is set up using PyCharm, where virtual environments and dependencies are configured. The YOLOv5 model is then executed using either CPU or GPU resources, depending on the system's hardware capacity. Integration is completed by linking the detection output to the Telegram alert system, and the functionality is tested using sample video feeds to ensure system accuracy and responsiveness.

4.2 Key System Features

The system delivers real-time video surveillance by continuously processing incoming frames with minimal delay, enabling swift detection of crowding situations. It maintains high detection accuracy even under suboptimal conditions, such as poor lighting or partial visibility of individuals. Alerts are automatically generated and dispatched with relevant details like timestamps and location identifiers to aid quick decision-making. Furthermore, the system provides flexibility through customizable crowd density thresholds that can be tuned for specific events or venues. Its design ensures efficient operation on standard hardware configurations, with optional GPU acceleration available to enhance performance for large-scale deployments.

4.3 Flow Chart

The figure below illustrates the logical workflow of the system:

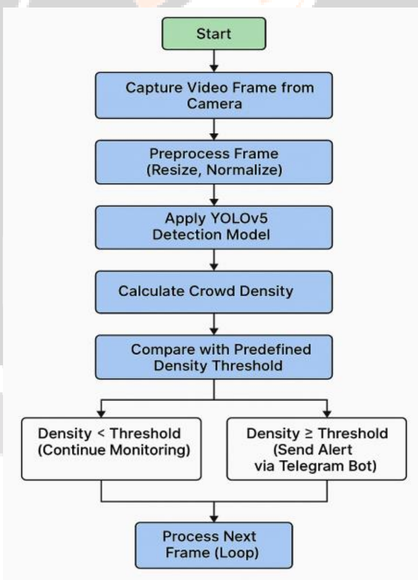

Fig3: Flow chart

5. Results and Discussions

The proposed real-time crowd density detection system was evaluated under various scenarios to assess its performance in practical environments. The testing conditions included low-density (fewer than 20 people), medium-density (20–50 people), and high-density (more than 50 people) crowds, as well as varying lighting conditions such as daylight, low light, and artificial illumination. The system was deployed on a setup comprising an Intel Core i5 (10th Gen) processor, 16 GB RAM, and an NVIDIA GTX 1650 GPU, using 1080p HD CCTV video feeds as input. The YOLOv5 model demonstrated high detection accuracy, consistently identifying individuals in low and medium-density scenes with over 90% accuracy. In high-density scenarios, minor accuracy drops were observed due to occlusion and overlapping, which affected detection precision.

The system maintained real-time responsiveness with an average processing rate of 25–30 frames per second, ensuring smooth live monitoring. Crowd density was calculated using a simple formula based on the number of detected individuals relative to the area monitored, which proved effective for estimating current conditions. Alerts were automatically generated through a Telegram bot when density exceeded a defined threshold, with minimal latency (typically under two seconds). While the system performed reliably under most conditions, slight reductions in accuracy occurred in low-light environments, and its reliance on GPU resources may limit deployment on edge devices with lower specifications. Importantly, the system does not incorporate predictive modeling for forecasting future crowd trends; it focuses solely on real-time detection and alerting. Future enhancements could involve integrating temporal models to add predictive capabilities, improve performance in poor lighting, and enable more proactive crowd control.

6.Conclusion

This project presents the development of a real-time system for detecting individuals and estimating crowd density using the YOLOv5 deep learning framework. The system captures live video feeds from surveillance cameras, detects and counts people, calculates crowd density, and triggers automatic alerts via a Telegram bot when predefined safety thresholds are exceeded. The model achieved high detection accuracy in low to medium-density scenarios and maintained real-time performance with minimal latency, proving suitable for environments that require continuous surveillance. By reducing human dependency and ensuring fast response times, the system contributes to safer crowd management in public spaces.

6.1 Future Research Directions

While the current implementation delivers reliable performance in both controlled and semi-controlled settings, several opportunities exist for future research and enhancement. Integrating advanced sensors such as thermal cameras, infrared detectors, or LiDAR could improve detection in low-visibility conditions like fog, darkness, or smoke, making the system more resilient for emergency and disaster response. Incorporating drone-based monitoring via UAVs could enable broader aerial surveillance, especially for large outdoor events where fixed cameras are insufficient. The addition of predictive analytics—leveraging historical data and machine learning—could allow authorities to forecast crowd surges and deploy resources preemptively. Multi-camera synchronization and data fusion techniques would expand system coverage, enhancing monitoring of large venues such as stadiums or airports. Furthermore, deploying the system on edge computing platforms (e.g., Jetson Nano, Raspberry Pi with NPU support) can reduce latency and reliance on cloud infrastructure, making it ideal for remote or bandwidth-constrained environments. Privacy protection must also be addressed through techniques like face anonymization, blurred detection overlays, and encrypted communication to comply with data regulations. Lastly, adaptive learning approaches, such as online learning and transfer learning, can help the model adjust to different crowd behaviors and environments, improving contextual accuracy over time.

7.References

[1] Gao, Wang, and Yuan proposed an attention-based regression model to enhance the precision of crowd estimation by focusing on both spatial and channel features within images.

[2] Erden et al. introduced a hybrid system combining infrared sensors and camera inputs for reliable people counting under various indoor conditions.

[3] Ryan and colleagues conducted a comparative study of multiple crowd estimation techniques, highlighting differences in feature extraction and model performance.

[4] Girshick et al. developed a hierarchical feature approach that significantly improved object detection and semantic segmentation tasks in complex environments.

[5] Marsden and team designed a fully convolutional network optimized for high-density crowd scenes, enabling more accurate headcount predictions.

[6] Yin, Velastin, and Davies explored the use of reference-based image processing to estimate crowd density, laying groundwork for earlier vision-based methods.

[7] Liu et al. utilized a deep learning architecture with recurrent spatial awareness to improve crowd counting accuracy across varying scenarios.

[8] Chen and colleagues introduced a region-focused approach for better localization and crowd analysis in complex scenes using predictive mapping.

[9] Benetka, Krumm, and Bennett examined contextual awareness in human-computer interaction, focusing on activity prediction.

[10] Bhangale et al. (2020) explored a deep learning-based approach for near real-time crowd counting, improving the efficiency and accuracy of counting individuals in crowded environments, as discussed in *Procedia Computer Science*.

[11] Yang et al. (2019) proposed a mobile system for counting people based on vision techniques, which allows for crowd monitoring on mobile devices, outlined in their paper at the 2019 4th International Conference on Machine Learning Technologies.

[12] Liu et al. (2021) introduced a crowd counting system based on people flow estimation, utilizing deep learning to model crowd movement, as discussed in *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

[13] Zhang and Chan (2019) introduced a wide-area crowd counting technique that combines ground-plane density maps and multi-view CNN fusion, providing an innovative solution for large-scale crowd monitoring, as presented at the IEEE/CVF Conference on Computer Vision and Pattern Recognition.

[14] Wang et al. (2020) developed a crowd counting method utilizing distribution matching, which optimizes the accuracy of people detection in crowded areas, shared in the *Advances in Neural Information Processing Systems*.

[15] Sindagi and Patel (2019) introduced a hierarchical attention-based network for crowd counting (Ha-CCN), focusing on improving the model's performance in challenging crowd situations, as outlined in *IEEE Transactions on Image Processing*.

[16] Yan et al. (2021) proposed a method for multi-domain crowd counting, enhancing the model's ability to generalize across different types of crowd scenarios, as detailed in *IEEE Transactions on Circuits and Systems for Video Technology*.

[17] Sam et al. (2020) introduced a method for accurately detecting and counting individuals in dense crowds, leveraging detection-based models for improved precision, as described in *IEEE Transactions on Pattern Analysis and Machine Intelligence*.