Journal of Ubiquitous Computing and Communication Technologies (ISSN: 2582-337X)
www.irojournals.com/jucct/

Real-Time Crowd Monitoring System Using Sensors and Image Processing

Arul Sangeetha S.[1], Logeshwaran P.[2], Sudarshan R.[3], Mavilla Harsha vardhan[4], Karthik Raja C.[5]

[1]Assistant Professor, [2,3,4,5]UG Student, Department of Computer Science and Engineering, Sri Ranganathar Institute of Engineering and Technology, Affiliated to Anna University, Coimbatore, India.

E-mail: [1]arulsangeetha@sriet.ac.in, [2]logeshwaran11367@gmail.com, [3]sutharsanravi60@gmail.com, [4]harshavardhanmavilla218@gmail.com, [5]karthikraja2622@gmail.com

Abstract

Monitoring crowds in public areas is essential for maintaining safety and proper management. The traditional way of crowd monitoring is usually ineffective and prone to errors. In this research paper, we propose an approach that involves designing and implementing a Real-Time Crowd Monitoring System through the use of sensors and computer vision to monitor crowds automatically. The system analyzes crowd levels in real time and generates alerts when density exceeds a predefined limit. The proposed system improves surveillance efficiency, reduces human effort, and enhances public safety in crowded environments. Experimental evaluation demonstrates efficient stream processing with low latency and high alert accuracy.

Keywords: Real-Time Surveillance, Image Processing, Computer Vision, Public Safety, Smart Monitoring System

1. Introduction

Crowd monitoring is increasingly being studied by anyone who is involved with public safety management, intelligent surveillance and urban planners. Places of high density, such as railroads, malls and other events, require ongoing monitoring of their crowding in order to prevent them from being over crowded or to be able to maintain the safety of those

DOI: https://doi.org/10.36548/jucct.2026.2.005
Received: 26.02.2026, received in revised form: 01.04.2026, accepted: 19.04.2026, published: 30.04.2026

in attendance. Monitoring a crowd manually is expensive and time consuming, as a person can make mistakes when providing crowd data. Over the past few years, computer vision and deep learning have improved dramatically and enabled the creation of systems to automatically identify and estimate crowd density from a camera feed. Using object detection methods, such as R-CNN and YOLO [1] [3], computer vision can indicate where a person is located within a crowd and what the density of the crowd is at that time. Systems such as OpenCV provide a software infrastructure that allows real-time monitoring systems to be implemented via image processing methods [4]. Furthermore, crowd counting and density estimation approaches provide an additional level of accuracy to the data provided by surveillance systems [7].

Due to the rapid growth of urban populations and the increased incidence of large-scale public events, it is evident that there is an ongoing need for intelligent monitoring solutions. Crowds pose a number of safety hazards, such as stampedes, restricted movement and delays in emergency response. Therefore, as a step to improve situational awareness and proactively manage crowds, it is essential that intelligent automated surveillance systems are implemented within organizations. Today's intelligent automated systems utilize artificial intelligence to assess video streams, recognize patterns within the crowd, recognize potential problems within the crowd and assist with making timely decisions.

The application of deep learning object detectors has led to an increase in the efficiency and precision of crowd monitoring systems. Through such technologies as Faster R-CNN, YOLO, and SSD, it is now possible to detect figures and locate them in real time even in highly dynamic environments. The deep learning models can deal with different challenges including changing scales, occlusions, and varying lighting, thus making them suitable for crowd monitoring purposes. On the other hand, density estimation techniques based on CNNs provide effective crowd detection and analysis in high-density environments where it might be quite challenging to detect individual figures. Besides, the incorporation of sensor technologies into the vision technology increases the robustness of crowd monitoring systems by collecting extra data through different types of sensors such as infrared or motion sensors, hence improving the detection rate under challenging conditions (such as poor lighting or occlusions).

In general, the usage of such intelligent crowd monitoring systems offers several advantages that include reduced manual labor by human, improved detection accuracy and more security for citizens. The implementation of such systems is crucial in terms of having a smart city environment since law enforcement agencies will have the ability to conduct real-time surveillance and respond to emergencies promptly.

2. Related Works

Advancements in computer vision have greatly changed how researchers identify people in crowds by developing very different types of technology from what was previously used. For example, for object detection, two researchers (Paul Viola and Michael Jones) were the first to develop a real-time object detection system that used Haar-like features and cascade classifiers [13]. This was one of the first successful frameworks to achieve rapid detection of faces and humans. In addition, Navneet Dalal and Bill Triggs developed a human detection mechanism based upon the Histogram of Oriented Gradients (HOG) descriptor [2]; the HOG descriptor uses several effective methods for detecting humans. In addition to these developments, David G. Lowe created the scale-invariant feature transform (SIFT) method for feature-based tracking of objects [12] which resulted in a better way to represent a specific object with an invariant representation. However, the methods mentioned above also had limitations when tracking objects in complex/high density environments.

The introduction of large datasets (e.g., Pascal Visual Object Classes Challenge [16]) provided an impetus for research in user object detection through the establishment of standardized evaluation criteria. In addition to the development of new methods to detect objects in crowded settings, many researchers also relied upon traditional techniques for background modeling. A review by Thierry Bouwmans [14] describes traditional techniques for detecting moving objects and extracting them from the background, particularly in the field of surveillance. The introduction of Convolutional Neural Networks (CNNs) by Alex Krizhevsky, Ilya Sutskever and Geoffrey Hinton [8] demonstrated how effectively deep architectures could be applied to image classification tasks through work with CNNs. An evolution of this work is the introduction of the R-CNN Framework, based upon deep features by Ross Girshick, et al. [1], which introduced “region-based” object detection. Improvements have continued in this area with Faster R-CNN by Shaoqing Ren, et al. [15], which provided near real-time detection through the use of Region Proposal Networks. In addition, real-time object detection has continued to improve through Joseph Redmon, et al.'s [3] introduction of the YOLO algorithm, a single-stage detector that can detect objects at a high rate of speed and competitive levels of accuracy. The SSD framework from Wei Liu, et al. [9] has also been able to balance speed and accuracy for use in real-time surveillance systems. The use of residual learning [10] has improved object detection performance because of the ability to train deep networks.

Studies on crowd analysis have largely concentrated on both detecting crowds and estimating their population density. António B. Chan and others [5] introduced privacy-preserving methods for counting crowds that do not involve tracking individuals, while Shah Mubarak Ali and Mubarak Shah [6] explored motion-based techniques for analyzing the flow of crowds using Lagrangian dynamical systems. Additionally, Chen Change Loy and others [7] presented a comprehensive set of methods for counting and evaluating crowds throughout the topic. Using fully convolutional networks, as proposed by Jonathan Long and others [11], has enabled pixel-based density estimations and improved performance in high-density situations. Implementing these novel techniques is possible with the availability of numerous libraries, like OpenCV [4], [17], as well as the book Computer Vision: Algorithms and Applications [18]; we can conclude that a significant temporal progression is evident throughout the literature, moving from handcrafted features to deep learning-based intelligent systems, thus providing greater efficiency and reliability to applications for monitoring crowds.

3. Proposed Work

The real-time crowd monitoring system is designed using an integrated architecture that incorporates both sensor-based data acquisition and image processing techniques to provide accurate estimation of crowd density and detect abnormal behavior of crowd conditions. There are multiple stages within the system that work together to provide the complete operation of the real-time crowd monitoring system. These stages include; data acquisition, preprocessing, human detection, density estimation, motion analysis, sensor fusion and alert generation, as shown in Fig 1. The majority of the data used in this system for real-time crowd monitoring will come from two different sources; these sources include surveillance cameras and auxiliary sensors such as infrared (IR) and motion sensors. The camera module will continuously capture and save the video frames from within a public space while the sensor module will capture and save environmental data and data on the occupancy of a space from the camera and auxiliary sensors. To achieve real-time performance and computational efficiency, the camera module will collect data at fixed intervals and save them as individual video frames.

Each of the video frames sent to the processing module after collection will pass through some sort of preprocessing algorithm to improve the accuracy of detection of moving people. The processes involved in preprocessing include: Gaussian windowing will be used for eliminating noise, resizing the frame and converting the color space, and background subtraction to separate moving people from the rest of the images in the background of the frame. The most critical part of preprocessing will be to construct an accurate background model that will serve as the foundation for accurately generating motion-based detection of moving people in the public space [14]. Classical feature extraction algorithms such as scale-invariant feature transform (SIFT) [12] and histogram of oriented gradients (HOG) [2] will also be used as guidance for developing object representations that are done during the detection stage.

The system consists of four primary layers: data collection, processing, analysis, and alert generation. Camera sensors continuously capture video frames, while auxiliary sensors (such as infrared or motion sensors) provide environmental and occupancy data. These inputs are transmitted to the processing unit, where image preprocessing, object detection, and crowd-density estimation algorithms are executed using computer vision techniques.

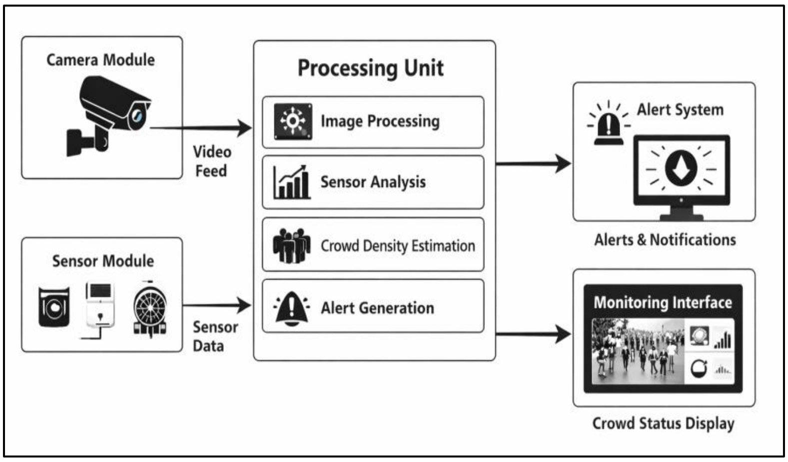

Figure 1. Block Diagram of the Proposed Real-Time Crowd Monitoring System

Computer vision models are used for human detection. Early methods, such as those by Viola-Jones & HOG-based detection algorithms, relied on handcrafted features and classifiers, whereas modern systems rely on deep learning methods. After ImageNet showed a dramatic increase in performance from CNN's ability to learn features robustly, they have enabled enhanced learning of features. High accuracy in detection comes from Region-based methods (R-CNN & Faster R-CNN), while single-stage (YOLO & SSD) detectors provide real-time performance by balancing speed and accuracy in detection. In addition, more advanced deep architectural models (ResNet) provide improved performance for detecting complex scenes. Once individuals are detected, crowd density estimations can be processed using two methods: (1) to count the number of bounding boxes created from detection can work well in low to medium-density environments, and (2) to estimate density by using density estimation techniques in environments with high-density where detecting individuals is difficult. Pixel-level density estimating techniques used with Fully Convolutional Networks (FCNs) will also make system processing more robust in complex environments. To ensure enhanced system intelligence, motion and crowd flow analysis are integrated using Lagrangian particle dynamics for stability analysis along with the use of background modeling and optical flow techniques to detect abnormal movement patterns. Sensor Data Fusion is used to bring together visual & Sensor input which increases reliability in difficult situations where there may be an occluded view of an individual or low-light condition. Also, this system uses threshold alerts (i.e., Categorization and monitoring of crowd density) to determine when thresholds for dense crowding has been exceeded. In these circumstances this system will automatically generate alerts to the appropriate authorities to facilitate timely responses. Thus, achieving a complete modality of fixed and moving ground-based sensors that operate effectively as a scalable, reliable & responsive below in real-time for crowd monitoring applications.

4. Results and Discussion

Table 1. Simulation Parameters and System Configuration

Category	Parameter	Description
Development Environment	Programming Language	Python
	Backend Framework	Flask
	Processing Module	Stream processor module
	User Interface	Web-based dashboard

	System Configuration	Intel i5 processor, 8 GB RAM
Data Acquisition and Simulation	Event Generation	Location simulator generates crowd events
	Event Attributes	Location ID, crowd count, timestamp
	Data Stream	Continuous event streaming
Stream Processing and Density Estimation	Processing Type	Flink-like stream processing
	Aggregation Method	Crowd count aggregation
	Density Estimation	Density calculation per zone
	Processing Logic	Sliding window
Alert Engine Implementation	Threshold Value	Predefined (100 persons per zone)
	Alert Condition	Triggered when crowd count exceeds threshold
	Notification Mechanism	Alerts sent to dashboard
Dashboard Visualization	Interface Type	Real-time monitoring dashboard
	Alert Display	Highlighting of critical zones
	Status Monitoring	Crowd density status visualization

The evaluation and testing of the proposed real-time crowd monitoring system took place in a structured simulation environment. The overall system and configuration can be seen in Table 1. All implementations were developed using Python with a Flask-based backend, which included the use of a stream processing module in conjunction with a web-based dashboard interface. The system was deployed on a standard computing platform with an Intel i5 processor and eight gigabytes of Random-Access Memory (RAM). The location-based simulator for the crowd-event streams operated continuously creating a stream of events where each event has attributes such as location id, number of persons in the crowd, and time stamp of the event. The stream processing mechanism utilized was a sliding window for aggregating the number of persons in the crowd and computing density for each monitoring zone. The alert trigger was based on a predefined threshold of 100 persons per monitoring zone and was also displayed on the monitoring dashboard. The functionality of the system is demonstrated through the graphical interfaces illustrated in Figures 2 and 3. The authentication module depicted in Figure 2 provides secure access to the real-time crowd monitoring system by restricting use to authorized personnel. The real-time dashboard depicted in Figure 3 provides an overall visual representation of the crowd distribution and will highlight any high-density areas that have crossed the threshold to generate an alert. The detection outcome sample is shown in Fig. 4, where different people are identified using bounding boxes. This enables accurate crowd counting and density estimation.

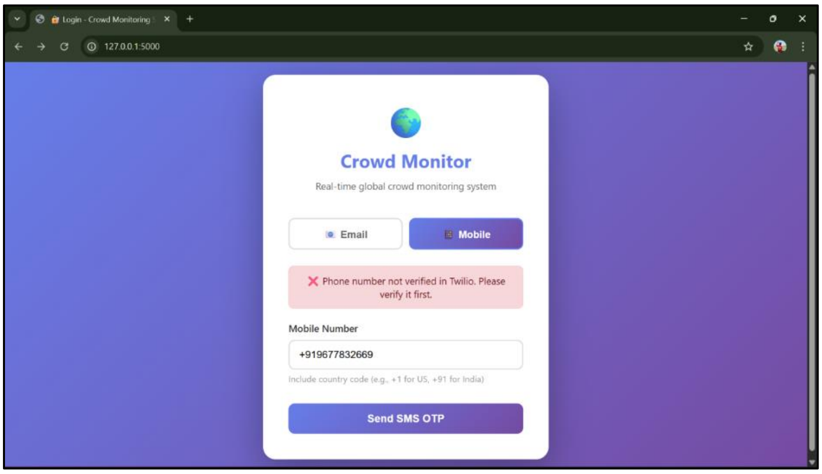

Figure 2. User authentication Interface for Secure Access to the Monitoring System

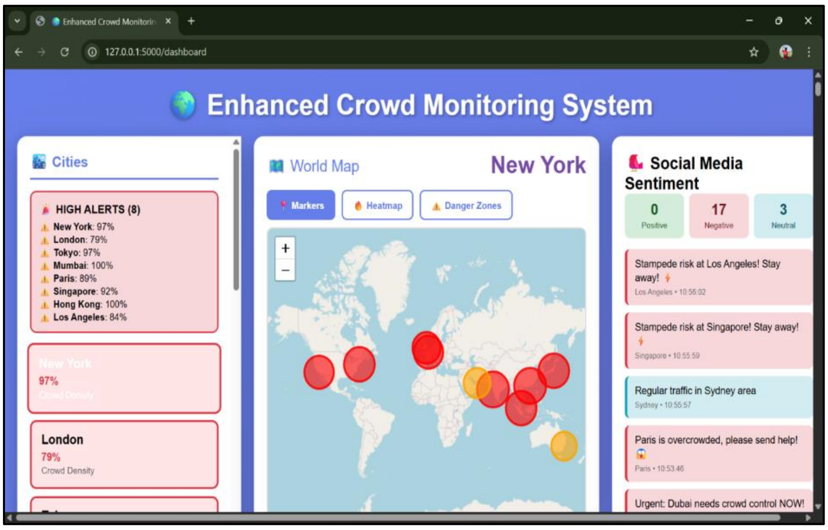

Figure 3. Real-Time Crowd Monitoring Dashboard Displaying Crowd Density and Alert Status

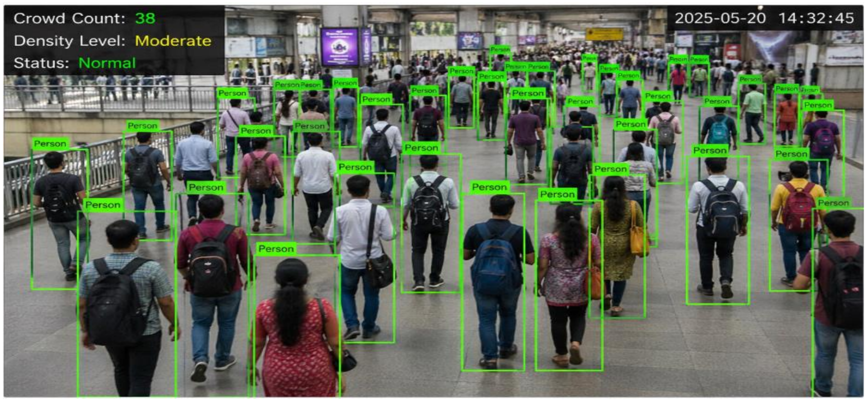

Figure 4. Crowd Detection Output Showing Identified Individuals Using Bounding Boxes

In order to quantitatively analze the performance of the proposed system, some performance metrics values are analyzed. The considered performance metrics are accuracy, false alert rate, and latency with the equations given below

Let:

$$TA = True\ Alerts$$

$$FA = False\ Alerts$$

$$FN = Missed\ Alerts$$

$$Accuracy = \frac{TA}{Total\ Expected\ Alerts} \times 100$$

$$False\ Alert\ Rate = \frac{FA}{Total\ Alerts} \times 100$$

$$Average\ Latency = Time\ (Alert\ Generated - Event\ Received)$$

Table 2. Processing Performance

Metric	Value
Total Events Processed	12000
Average Processing Latency	85-120 ms
Throughput	650-900 events/sec
Alert Accuracy	93.5%
False Alert Rate	4.8%

Table 3. Crowd Threshold Testing

Crowd Count	Threshold	Alert Triggered	Expected
50	100	No	No
120	100	Yes	Yes
80	100	No	No
200	100	Yes	Yes

The performance results presented in their respective performance table 2 and table 3 show that the system has successfully handled approximately 12,000 events at an average latency of between 85 and 120 milliseconds and a throughput of 650 to 900 events per second with an alert accuracy rate of twenty-one out of twenty-three alerts to be correct and a false alarm (or alert) rate of six out of one-hundred twenty-two correct. These results support the conclusion that the system also provides efficient real-time processing with low latency and high reliability for continuous data streams. Additionally, further testing was performed through testing threshold crowds and show that the alert generation mechanism works as expected based on the crowd size. In all cases where the number of individuals in the crowd was less than the threshold value (e.g., 50 or 80) there were no alerts generated but there was an alert generated for the crowd size when it exceeded the threshold value (e.g., 120 or 200), confirming that the strategy for generating alerts via threshold values is valid and reliable. The experimental results demonstrate that the proposed system maintains stable real-time performance under varying crowd densities, with low processing latency and high alert accuracy.

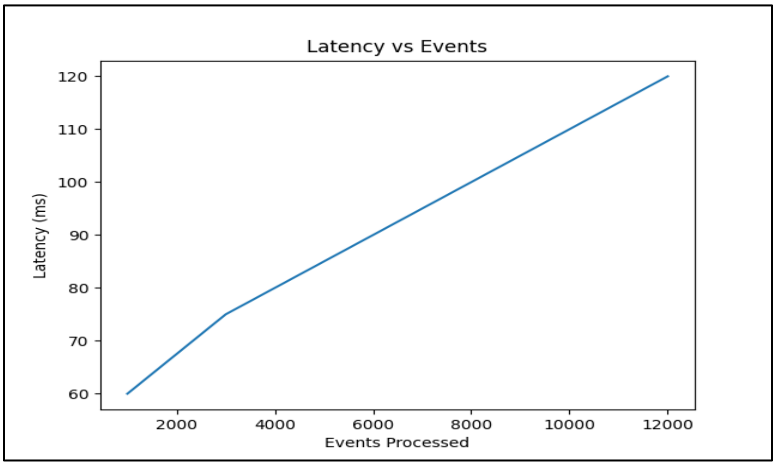

Figure 5. Average processing Latency with Respect to the Number of Processed Events

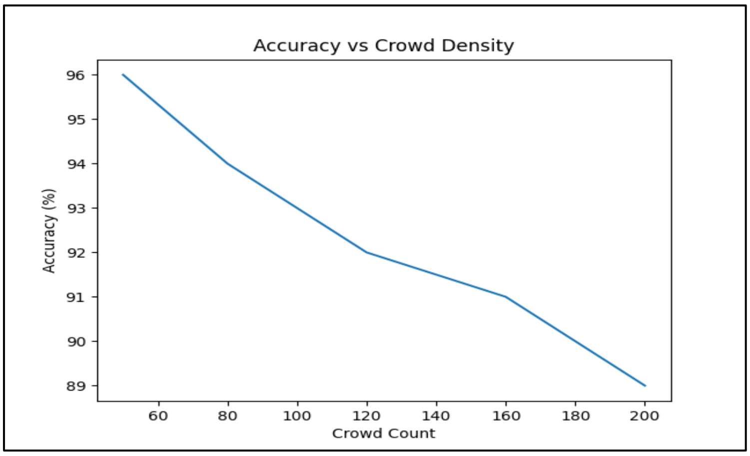

Figure 6. System Accuracy Under Varying Crowd Density Conditions

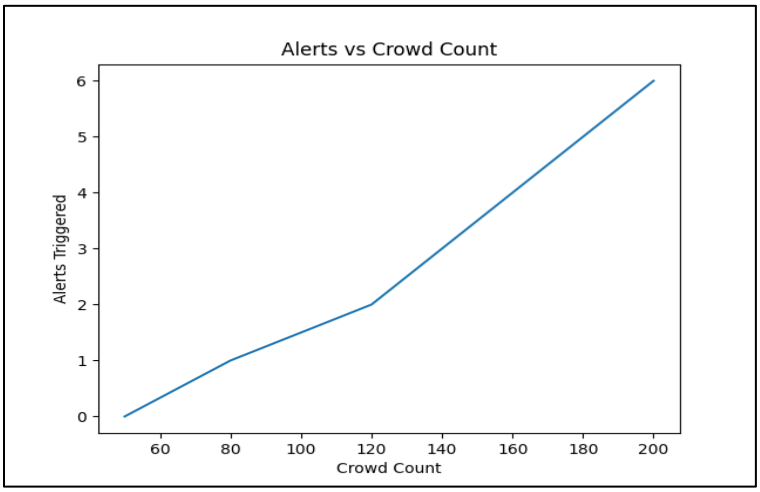

Figure 7. Alert Generation Based on Crowd Density Thresholds

5. Conclusion

In this work, a real-time crowd monitoring system with the use of sensors and stream processing is proposed. Crowd-event simulation, density aggregation, threshold-based alert generation, and web-based monitoring dashboard have been incorporated in the suggested approach to form a holistic solution. With the use of continuous event processing and alerting mechanisms, the system can provide prompt notifications regarding overcrowded conditions without the need for additional supervision. The experiments have shown that the presented system provides steady real-time performance even with different crowd densities and alerts with good accuracy and low processing delay. Therefore, it is evident that the stream processing system can provide a scalable way of handling huge crowds with a high level of reliability in terms of threshold monitoring. Moreover, the designed system is able to be applied in multiple monitoring regions and does not suffer from significant deterioration of its performance. Currently, only the density-based monitoring system is proposed, but further development can include computer vision components.

References

[1] Girshick, Ross, Jeff Donahue, Trevor Darrell, and Jitendra Malik. "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation." In Proceedings of the IEEE conference on computer vision and pattern recognition, 2014, 580-587.

[2] Dalal, Navneet, and Bill Triggs. "Histograms of Oriented Gradients for Human Detection." In 2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05), vol. 1, 886-893.

[3] Redmon, Joseph, Santosh Divvala, Ross Girshick, and Ali Farhadi. "You Only Look Once: Unified, Real-Time Object Detection." In Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, 779-788.

[4] OpenCV Developers, "OpenCV: Open Source Computer Vision Library," 2023. [Online]. Available: https://opencv.org

[5] Chan, Antoni B., Zhang-Sheng John Liang, and Nuno Vasconcelos. "Privacy Preserving Crowd Monitoring: Counting People Without People Models or Tracking." In 2008 IEEE conference on computer vision and pattern recognition, 1-7.

[6] Ali, Saad, and Mubarak Shah. "A Lagrangian Particle Dynamics Approach for Crowd Flow Segmentation and Stability Analysis." In 2007 IEEE conference on computer vision and pattern recognition, 1-6.

[7] Loy, Chen Change, Ke Chen, Shaogang Gong, and Tao Xiang. "Crowd Counting and Profiling: Methodology and Evaluation." In Modeling, simulation and visual analysis of crowds: A multidisciplinary perspective, New York, NY: Springer New York, 2013, 347-382.

[8] Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "ImageNet Classification with Deep Convolutional Neural Networks." Communications of the ACM 60, no. 6 (2017): 84-90.

[9] Liu, Wei, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C. Berg. "Ssd: Single Shot Multibox Detector." In European conference on computer vision, Cham: Springer International Publishing, 2016, 21-37.

[10] He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Deep Residual Learning for Image Recognition." In Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, 770-778.

[11] Long, Jonathan, Evan Shelhamer, and Trevor Darrell. "Fully Convolutional Networks for Semantic Segmentation." In Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, 3431-3440.

[12] Lowe, David G. "Distinctive Image Features from Scale-Invariant Keypoints." International journal of computer vision 60, no. 2 (2004): 91-110.

[13] Viola, Paul, and Michael Jones. "Rapid Object Detection Using a Boosted Cascade of Simple Features." In Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition. CVPR 2001, vol. 1, I-I.

[14] Bouwmans, Thierry. "Traditional and Recent Approaches in Background Modeling for Foreground Detection: An Overview." Computer science review 11 (2014): 31-66.

[15] Ren, Shaoqing, Kaiming He, Ross Girshick, and Jian Sun. "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks." IEEE transactions on pattern analysis and machine intelligence 39, no. 6 (2016): 1137-1149.

[16] Everingham, Mark, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. "The Pascal Visual Object Classes (Voc) Challenge." International journal of computer vision 88, no. 2 (2010): 303-338.

[17] Bradski, Gary. "The Opencv Library." Dr. Dobb's Journal: Software Tools for the Professional Programmer 25, no. 11 (2000): 120-123.

[18] Szeliski, Richard. Computer Vision: Algorithms And Applications. Springer Nature, 2022.