



Original Research Paper

REAL-TIME MULTI-THREADED BASED OBJECT DETECTION AND COUNTING SYSTEM USING YOLOV8 AND BYTETRACK

S Vimal¹, M. Kaliappan², Gaurav Dhiman^{3,4,5}

¹Department of Computer Science and Engineering, KIT- Kalaingar Karunanidhi Institute of Technology, Coimbatore, Tamil Nadu 641402 India

²Department of Artificial Intelligence and Data Science, Ramco Institute of Technology, Rajapalayam, Tamil Nadu 626117, India

³Department of Computer Science and Engineering, Yuan Ze University, Tao Yuan, Taiwan

⁴University Centre for Research and Development, Chandigarh University, Mohali, India

⁵Division of Research and Development, Lovely Professional University, Phagwara, India

Email: kaliappan@ritrjpm.ac.in, svimal@kitcbe.ac.in, gdhiman0001@gmail.com

Abstract

Object detection from real-time video streaming is a challenging task, and this paper presents an innovative approach for a real-time object detection and counting system with YOLOv8. For the multi-object tracking from the detected objects of YOLOv8, ByteTrack is used. The designed system is used for long-duration video streams of about two hours of continuous video segments extracted from the static surveillance feeds. The multithreaded architecture utilized here is to read the frame with maximized throughput, with mitigated frame loss. Meanwhile, the object counting is performed with the user-defined virtual line with reduced jitter using the bounding box smoothing. The duplication of object counting is avoided with the inclusion of an intelligent ID system. The object bounding boxes, live count overlay, and center points are added to the disk in the real-time processing for the robust video writing approaches. To avoid resolution problems, the custom re-encoded pipeline is used. This can be used in real-time applications including industrial automation, retail analytics, and traffic monitoring.

Keywords: Object detection, YOLOv8, Byte Track, and Frame Grabber Thread

1. Introduction:

Computer vision is used by item identification and counting devices to recognize and tally elements in images or video sources. Usually, object detection, which finds items in the image, is accompanied by monitoring and counting to figure out how many of those objects there are. Because roads enable accessibility to economic development and transportation between locations, roads are essential to everyday life. But in order to be used to their full potential over time, highways need to be regularly maintained and repaired. From the state of the road structure to the

***Corresponding Author:**

Received: 25th July, 2026 | Revised: 26th August, 2026 | Accepted: 18th September, 2026 | Available Online: 24th September, 2026

(DOI): 10.70102/AEJ.2026.18.8s.3

nature of the surroundings, a number of factors must be taken into account when deciding how servicing and repairs are carried out. To determine which road segments require service and repairs, an ADT might be utilized.

Applications for this sort of device can be found in a number of industries, such as production, sales, traffic control, and transportation [1]. It locates and identifies features in an image using techniques such as CNNs or YOLO (You Only Look Once). Bounding boxes are frequently created by these techniques surrounding every object they identify. In a video platform, surveillance makes certain that every object that is recognized is tracked between images. This gives a more precise estimation and allows for multiple counting, particularly when working with moving components. The applications are manufacturing, retail, traffic management, logistics, and security [2]. Conveyor belt object counting can be automated to increase productivity and lower error rates in manufacturing. On-site client surveillance and counting can yield useful information for improving inventory control, personnel, and shop designs. Traffic circulation can be tracked, congestion can be found, and roadway signals can be optimized via object identification and counting.

Transportation and handling inventory can be made more efficient by counting boxes and parcels in storage. Object identification and counting in monitoring film can improve safety procedures and identify irregularities [3]. By automating laborious counting activities, object identification and counting systems may save staff resources. It can provide immediate information by analyzing information precisely and swiftly. Devices can be expanded to accommodate complicated situations and high data quantities. It saves energy and time by replacing laborious counting, beneficial for traffic flow analytics and tracking, with restructuring, effective in various fields, and high precision using contemporary deep learning methods [4]. A substantial amount of memory and processing power may be needed for the operation and execution of such networks. Dim lighting, object overlap, and occlusions can all have an impact on efficiency. To develop deep learning algorithms, massive collections of labeled images are needed, which might take a lot of effort to gather and interpret.

The rest of the work is organized as follows: the literature review is added in section 2, and the proposed methodology is elaborated in section 3. The results and discussion section is added in section 4. Finally, the work is concluded in section 5.

2. Literature survey:

Road segment maintenance and repair depend on the average daily traffic survey (ADT). This technique often employs a semi-manual technique and utilizes CCTV to count automobiles. Bihanda et al. [5] presented a Real-Time Detection Transformer (RT-DETR) structure for vehicle recognition and ByteTrack for monitoring and calculating in an ADT. A multi-vehicle voting mechanism was suggested to prevent an identical automobile from being mistakenly identified during the monitoring procedure before the vehicle determination. To assess the efficiency and reliability CCTV camera footage from various scenes was analyzed. The suggested technique was faster, high precision in crowded scenes, and complete, anchor-free identification. However, the

efficiency could suffer in bad conditions.

Due to their exceptional performance in several computer vision applications, Wang et al. [6] described the YOLOv8 series systems have emerged in numerous domains to improve their identification capabilities for small objects with the Symmetric Positive Definite Convolution (SPD-Conv) component. To attain higher accuracy, the sensor was integrated with the ByteTrack tracking approach, and the loss function and the IoU were improved. It was effective in real-time application and helps to handle missing identifications. Hence, it was not the best option for dense scenarios.

Automated transport administration relies heavily on the use of CCTV cameras for vehicle recognition, authentication, and automatic counting. Akoushideh et al. [7] developed deep convolutional neural networks (CNNs) for video monitoring for vehicle recognition, categorization, and counting were examined. The suggested system makes use of the ByteTrack method for following vehicles between frames. The YOLO framework for identifying objects and recognizes the Multiple-Zone-Line technique to tackle monitoring issues. The suggested approach presented a viable approach for reliable and precise vehicle recognition, categorization, and counting. The suggested method was able to distinguish between different car sorts and dimensions. However, it is simple to misclassify vehicles that have equivalent looks, which might result in inaccurate group counts.

The need for effective traffic detection and monitoring systems has increased in tandem with the swift rise in transportation-related stress. Liu et al. [8] established YOLOv8n to minimize the estimation of redundant parameters to minimize the estimation of redundant parameters implement SCC_Detect using SCConv to identify components. The Focaler-ElIoU loss function increased the precision and consistent monitoring and identification. It was simple to adjust for varying video quality. Hence, not the best option for lightweight devices, such as edge systems that lack compression or reduction capabilities.

Video-based traffic surveillance devices and CCTV camera image processing have advanced significantly due to the quick development of deep learning and computing power. Mutlukaya et al. [9] developed an automated traffic monitoring technique that uses strong deep-learning algorithms to provide an adequate structure for effective traffic detection. The system presented several significant features, such as a combined technique for complete traffic rule violation identification and an original technique for lane identity (LaneID) assessment. To successfully determine lane breaches, the investigation uses a variety of cutting-edge techniques, including DeepSORT, YOLOv8, and HybridNets. The method was excellent for implementation in low-latency real-world traffic networks and makes use of flexible, effective designs that strike a balance between effectiveness and precision [13]-[25]. Security concerns are brought up by real-time monitoring, particularly when combined with individual identification or registration number detection [26]-[34].

3. Proposed Methodology

The proposed methodology is to detect the cement bags using YOLOv8. For this, the feeds from the camera are captured frame by frame using the Frame Grabber Thread (FGT) with the predefined sampling rate. The objects from the captured frames are detected using YOLOv8, and the final decision for the detection is performed using ByteTrack, which uses multi-object tracking. The number of objects that are detected using the Counting Logic Thread (CLT) is displayed in the display thread. The full pipeline is depicted in Figure 1.

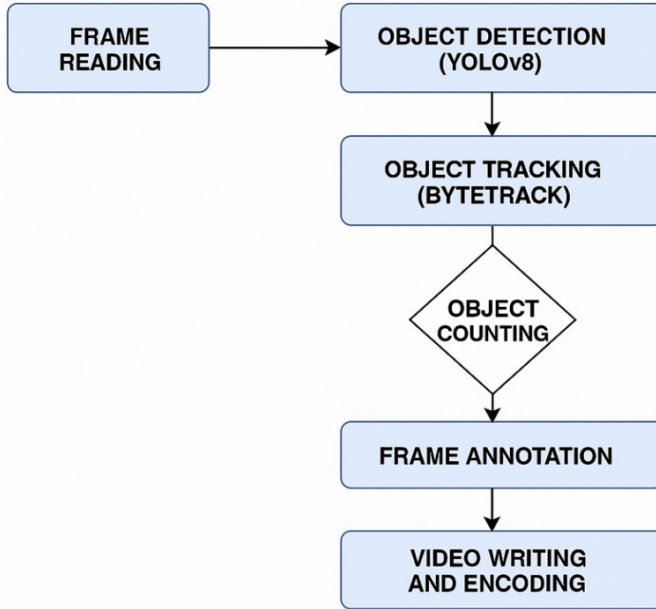


Fig 1: Pipeline of the proposed YOLOv8 based cement bag detection.

3.1. FGT for capturing the frames

FGT is the real-time video sampling mechanism and is used to capture the frame-to-frame from the feed that is live captured using the camera. This might have been performed using a predefined sampling rate [10]. The decoupling of video captured using the detection and the pipeline tracking for maintaining constant, smooth, real-time processing. This follows multithreading. Let us consider, at time t , the video frame is $VF(t)$ with the frame rate of vf_r .

The period between the consecutive frames is ∇t which is equal to $\frac{1}{vf_r}$. The buffer of the frame queue is implied as B_{vf} and the grabbing time of the frame is T_{Gr} . Based on the frame grabber, the sampling of the video feed can be formulated as,

$$VF_m = VF(t_0 + m \cdot \nabla t), \quad m \in M \quad (1)$$

The m th captured frame is described as VF_m with the initial time stamp of t_0 . The pushing of each frame to the queue can be formulated as,

$$B_{vf} \leftarrow VF_m \text{ if } T_{Gr} < \nabla t \quad (2)$$

(3)

The grabbing thread is executed asynchronously for the processing of real-time values and the frames are buffered using the following eqn,

$$Throughput_{Gr} \geq Throughput_{Dr}$$

The bottlenecks are averted with the decoupling of the frame obtained during the object detection. To avoid the frame lags and to maintain the real-time consistency,

$$T_{Gr} + T_P < \nabla t \quad (4)$$

Here, the time taken to grab the frame is defined as T_{Gr} with the processing time of T_P . Meanwhile, lagging frame can be implied as,

$$T_{Gr} + T_P > \nabla t \Rightarrow \text{lagging of frames} \quad (5)$$

The decoupling of grabbing can be performed with the consideration of $Thread_1$ and $Thread_2$ for the grabbing, handling, and detection tracking. The queue occurs during the communication is Bc_{vf} . The FIFO models for thread safety is presented as,

$$Bc_{vf}(t) = \{VF_m, VF_{m+1}, \dots, VF_{m+l}\}, \quad l \leq \text{max size of Queue} \quad (6)$$

3.2 YOLOv8-based detection thread

The detection of objects from each frame is effected with the proposed YOLOv8 from the frames of Bc_{vf} . The main aim of the YOLOv8 is to identify the bounding boxes for the objects from the frames and transmit the outcomes to the next stage, which is used for tracking the objects [11]. The input frame is considered as,

$$VF_m \in R^{h \times \omega \times 3} \quad (7)$$

The height of the frame is h and the width is ω and the color spacing is RGB. The set of detected output obtains for the frame VF_m can be formulated as,

$$OD_m = \{(y_j, z_j, W_j, H_j, C_j, P_j) | j = 1, 2, \dots, M_D\} \quad (8)$$

The center of the bounding boxes are (y_j, z_j) with the height and width of (H_j, W_j) . The class ID is (C_j) with the confidence score of P_j . The number of objects detected per frame F_m . For estimating the inference time constraints, assume the detection time per frame is T_D with the deadline interval of $T_{Deadline} = \nabla t$. While considering real-time processing, it should be, $T_D \leq \nabla t$, otherwise, it lags behind the frame grabber. The placing of outcomes in the detection queue can be performed using the following equation,

$$B_D \leftarrow (VF_m, d_m) \quad (9)$$

This queue is transmitted to the tracking thread.

ByteTrack-based thread tracking

In this section, the output obtained from the YOLOv8 OD_m can be used for multi-object tracking. This might have maintained the identities of frames. Assume that the previously tracked objects at time t_{m-1} be considered as,

$$T_{m-1} = \{ \kappa_1^{m-1}, \kappa_2^{m-1}, \dots, \kappa_l^{m-1} \} \quad (10)$$

The state for each track κ_j can be identified as,

$$\kappa_j = (y, z, W, H, u_y, u_z, ID) \quad (11)$$

The ID and the velocity of each object is determined as, ID and (u_y, u_z) . The updation of tracks is done with the association of detection output OD_m ,

$$IoU(\kappa_j, D_j) = \frac{area(\kappa_j \cap D_j)}{area(\kappa_j \cup D_j)} \quad (12)$$

This enhances the high-confidence detection first, with the ignorance of low-confidence detections. For low confidence, the re-matching can be obtained with the Kalman filter predictions [12]. The status of the tracks is updated with the Kalman filter as shown below,

$$\tilde{y}_m = E\tilde{y}_{m-1} + Fv_m + W_m \quad (13)$$

$$x_m = OB\tilde{y}_m + u_m \quad (14)$$

Here, the transition matrix is E with the F control matrix. The observation matrix OB can be processed, and the noises obtained are W_m and u_m . The tracking outputs are defined as,

$$T_m = \{(y_j, z_j, W_j, H_j, ID_j)\} \quad (15)$$

The output is forwarded to the counter/display thread as follows,

$$Bc_t \leftarrow (VF_m, T_m) \quad (16)$$

3.4 Counting Logic thread

Spatial rules are applied to track the counting of objects as they pass through predefined areas. For this, the counting logic thread is used and identification is performed using the unique Track ID provided throughout the frames. The input obtained from the tracking thread can be defined as,

$$T_m = \left\{ (y_j, z_j, W_j, H_j, ID_j) \mid j=1,2,\dots,M \right\} \quad (17)$$

The estimation of the centroid can be performed using the following equation,

$$c_j = (y_j, z_j) = \left(y_j + \frac{W_j}{2}, z_j + \frac{H_j}{2} \right) \quad (18)$$

The trajectory dictionary is maintained using,

$$Trajec[ID_j] = [c_j^{m-1}, c_j^m] \quad (19)$$

While the counting line crossing condition is applied to track the objects crossing the top to bottom or from bottom to top. Based on that, the counting is incremented to,

$$Count_{up/down} = Count_{up/down} + 1 \quad (20)$$

To avoid the counting of objects several times, keep the $Set < ID >$ for the counted object. Count the objects if that are $ID_j \notin Counted_IDs$ and added to the count. The count output is displayed.

4. Results and Discussion:

This section explains the evaluation of proposed work using various evaluation measures and its results. Required the annotated video sequence tracking and detection with the suitable dataset to compute the proposed system performance. Figure 2 shows the sample dataset images based on cement bag.

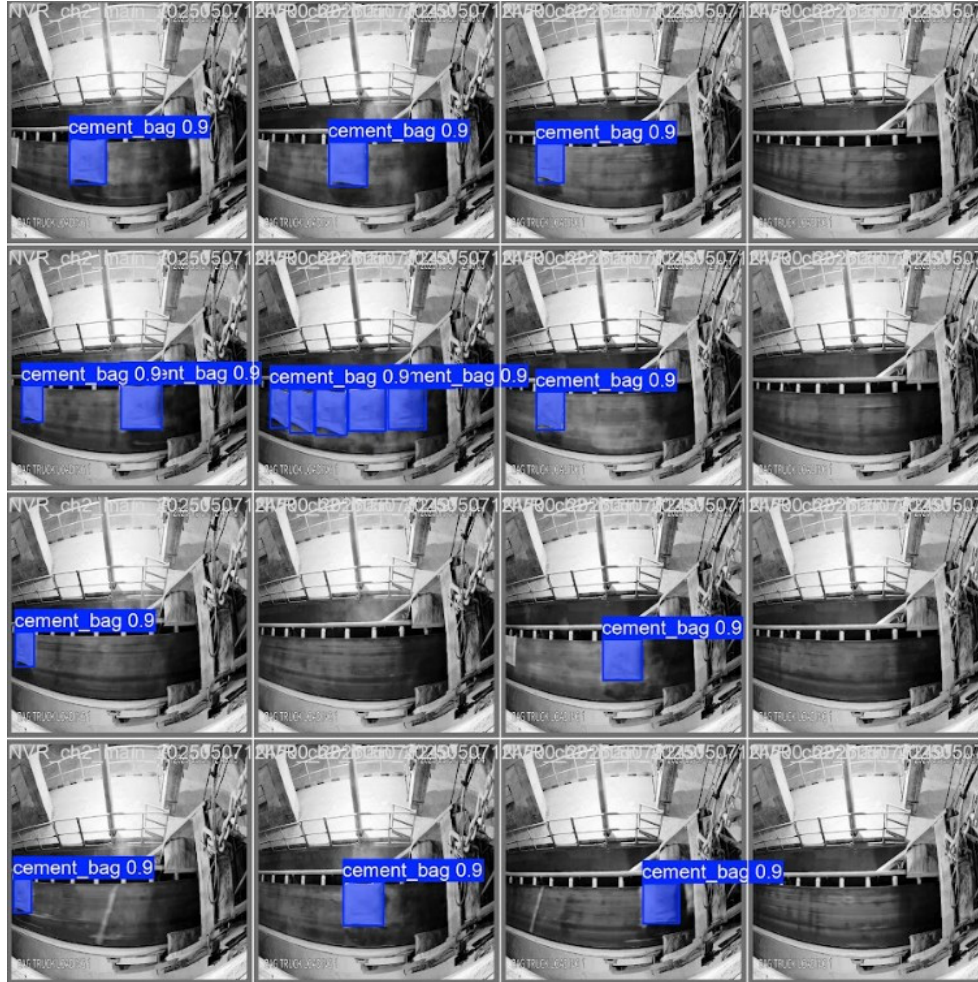


Figure 2: Sample dataset images

Figure 3 displays the performance validation curve along with the F1, precision and recall confidence levels are outlined in Figure 1 (i) to (iii). To assess the model performance, few critical measured are utilized namely F1, precision and recall confidence levels in object tracking and detection tasks. For real time object determination, the proposed object model with confidence levels increased from 0 to 1.0 respectively. Both comment bag and all classes result are above 95% when compared to the F1, precision and recall confidence levels.

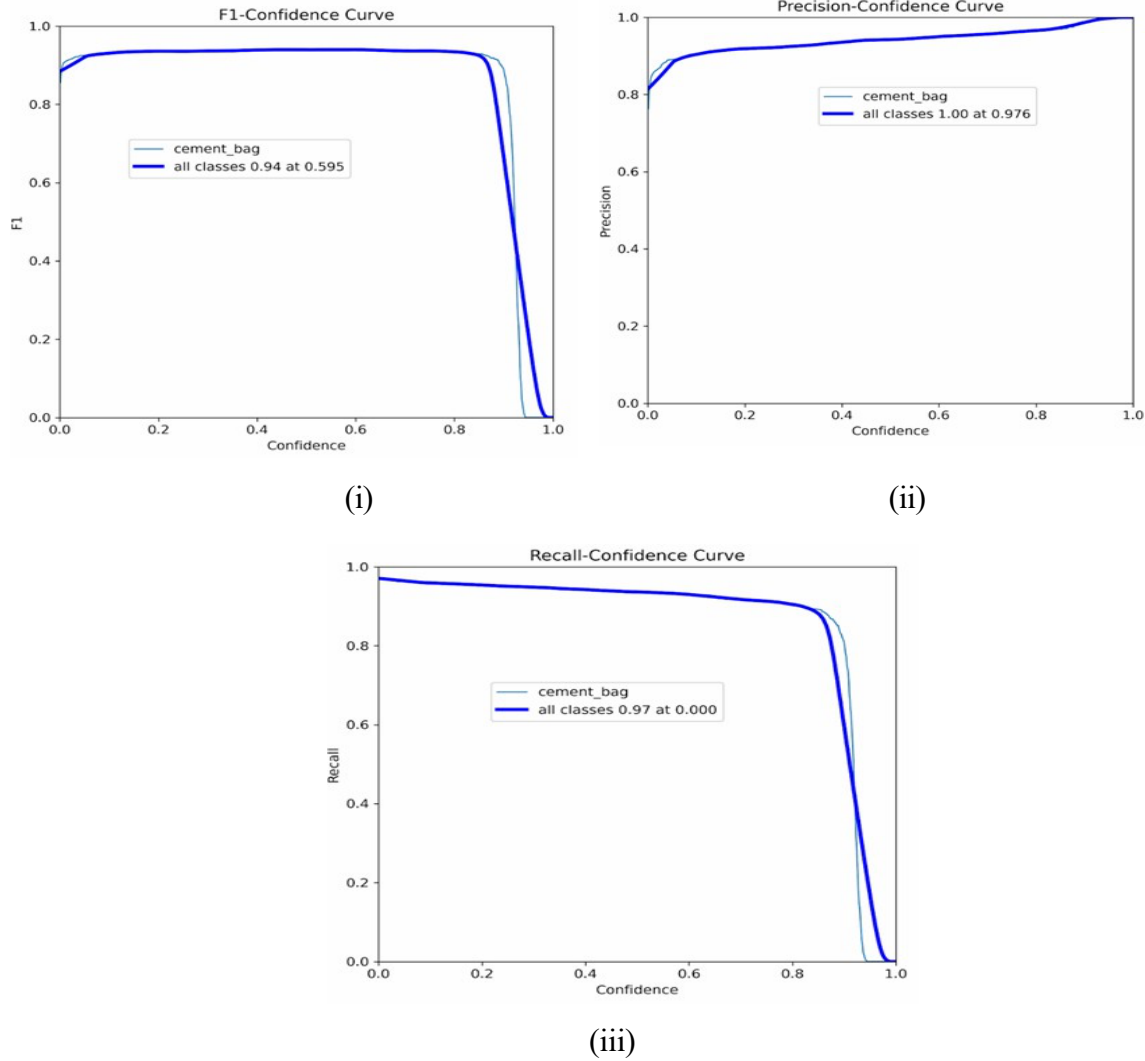


Figure 3: The performance evaluation outputs for, (i) F1-confidence level, (ii) precision level and (iii) recall level

Figure 4 delineates the confusion matrix results. The classification model performance for object detection measured using the key analysis tool is called confusion matrix. Each detection included object localizing and classification based on the bounding box although the detection not purely classification. An incorrect detection avoided as well as how better model differentiates among the object classes are assessed using confusion matrix.

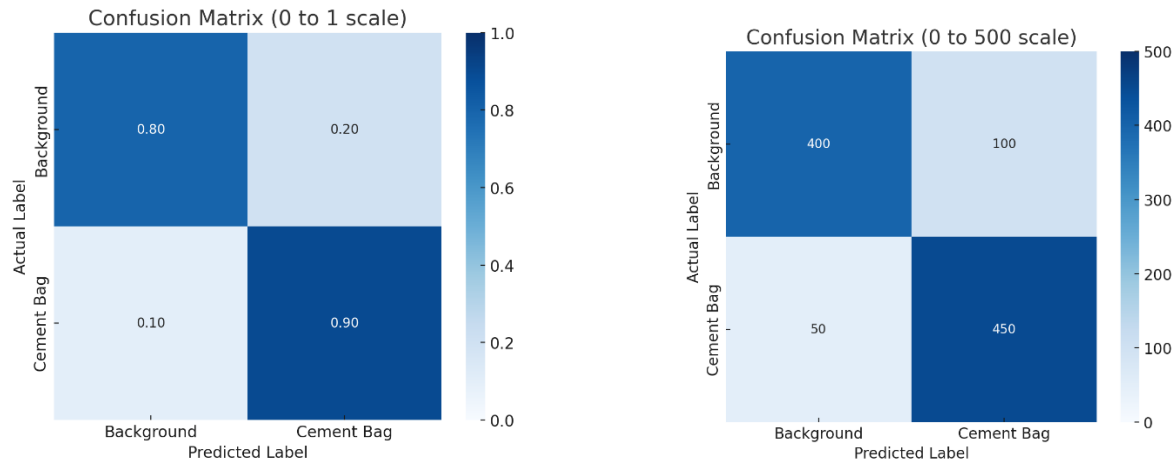


Figure 4: Confusion matrix results

The performance based on the labels curve is plotted in Figure 5. The bounding box and class ID coordinated with the label is video or image frame for each object in object detection tasks. Based on the image dimensions, the normalized with the x-centre, y-centre, height and width of four values defines the bounding box. The bounding box coordinates denoted using x and y centre. The image height and width divides the object corner bottom right and top left among the mid points are calculated. The object bounding box edges called x and y coordinates computed and the bounding box size represents the height and width.

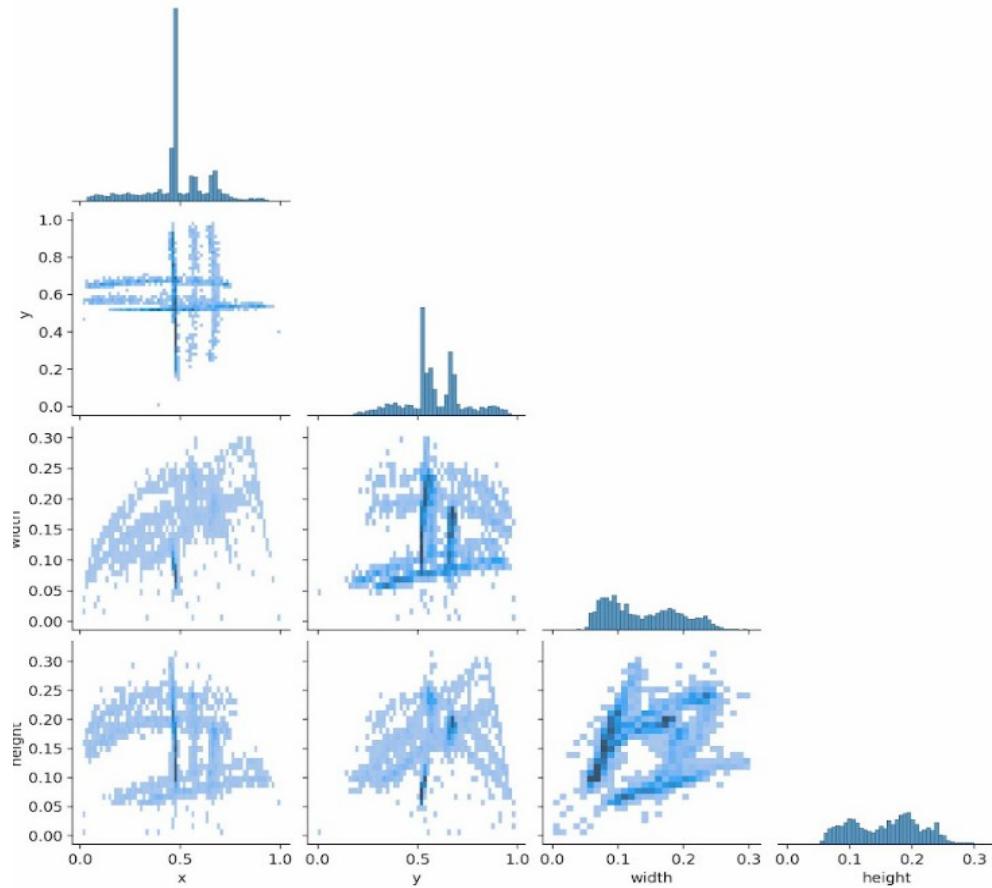


Figure 5: Performance based on the labels

Figure 6 outlines the mask performance for various measures like F-1, precision and recall curves are plotted in Figure 6 (i) to (iii). Across various models, the segmentation mask reliability and accuracy were evaluated with the proposed mask performance. To create an accurate segmentation mask, learned the mask model is learned steadily, increasing because of the imbalance between false negatives and false positives. Above 80% results were accomplished by both the cement bag and all classes for each measure. An optimal model performance is indicated as the peak in F1, precision, and recall results.

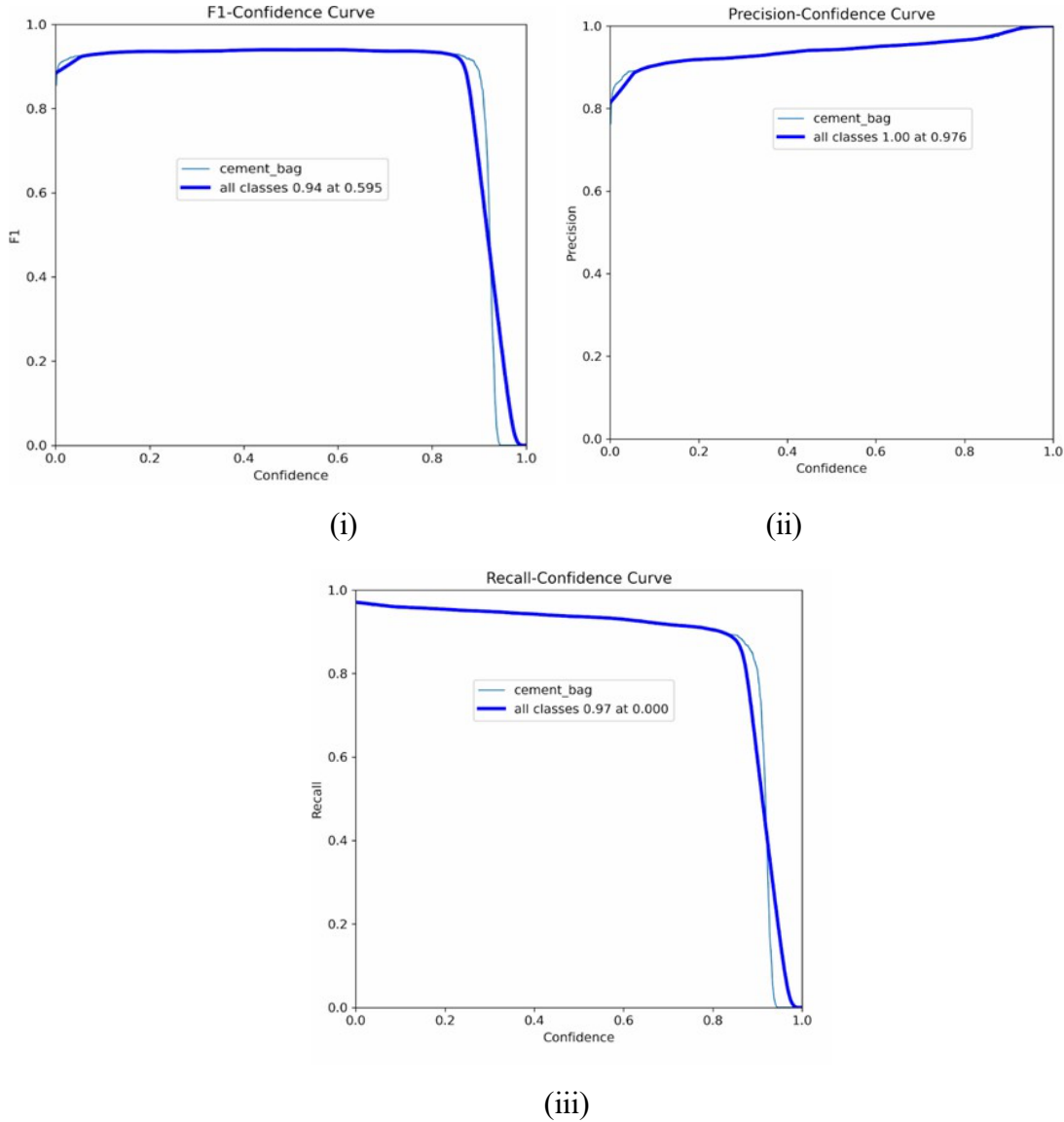


Figure 6: Proposed mask performance for (i) F1 curve, (ii) precision confidence and (iii) Recall curve

5. Conclusion

The implemented YOLOv8 and ByteTrack-based object counting system demonstrates an effective solution for real-time monitoring and video archival of long-duration footage. With intelligent tracking, accurate object counting logic, and robust handling of system resources through multithreading and queuing, the system maintains high accuracy even during frame bursts or processing lags. The integration of user-interactive ROI (region of interest) selection and temporal ID-based logic enhances flexibility and robustness across various surveillance scenarios. The output re-encoding mechanism ensures manageable file sizes without sacrificing detection quality, making this system deployable for long-term, large-scale surveillance needs.

Future enhancements could include adaptive line placement using scene context, cloud-based storage integration, and incorporation of deep learning-based re-identification for cross-camera tracking.

Acknowledgement: The authors are very grateful to the anonymous reviewers, associate editor, and editor for their insightful and constructive suggestions. The research is implemented in cement industry.

Funding Statement: This work is not financially supported.

Author Contributions: **C. Karpagavalli:** Writing – review & editing, Writing – original draft, Formal analysis, Conceptualization. **M. Kaliappan:** Writing – review & editing, Visualization, Supervision, Methodology, Conceptualization, Formal analysis. **V.Manimaran:** Supervision, Conceptualization, Formal analysis.

Availability of Data: Data will be made available on request.

Declaration of competing interest: The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References:

1. M Sivaram, M Kaliappan, SJ Shobana, MV Prakash, V PorkodiSecure storage allocation scheme using fuzzy based heuristic algorithm for cloud, *Journal of Ambient Intelligence and Humanized Computing*, pp.1-9.
2. M Kaliappan, E Mariappan, MV Prakash, B Paramasivan, Load Balanced Clustering Technique in MANET using Genetic Algorithms.. *Defence Science Journal* 66 (3), 251-258
3. S Vimal, YH Robinson, M Kaliappan, K Vijayalakshmi, S Seo , A method of progression detection for glaucoma using K-means and the GLCM algorithm toward smart medical prediction. *The Journal of Supercomputing*, PP.1-17
4. GS Kumar, M Kaliappan, LJ Julius, Enhancing the Performance of MANET using EESCP, *Pattern Recognition, Informatics and Medical Engineering (PRIME)*, 2012
5. Bihanda, Y.G., Fatichah, C. and Yuniarti, A., 2024. Multi-Vehicle Tracking and Counting Framework in Average Daily Traffic Survey Using RT-DETR and ByteTrack. *IEEE Access*.
6. Wang, Y. and Mariano, V.Y., 2024. A Multi Object Tracking Framework Based on YOLOv8s and Bytetrack Algorithm. *IEEE Access*.
7. Akoushideh, A., Sadat, S.S. and Shahbahrami, A., 2024. Counting vehicles types using deep learning algorithm in video surveillance systems. *Multimedia Tools and Applications*, pp.1-24.
8. Liu, Y. and Shen, S., 2025. Vehicle detection and tracking based on improved YOLOv8. *IEEE Access*.

9. Mutlukaya, I., Can Karakurt, R., Cetinkaya, S. and Bayraktar, E., 2024. Visual detection and tracking of lane violations for heavy vehicles. *Neural Computing and Applications*, 36(36), pp.22633-22652.
10. Trávníček, Zdeněk. "Multi-Threaded Real-Time Video Grabber." (2010).
11. Wang, G., Chen, Y., An, P., Hong, H., Hu, J. and Huang, T., 2023. UAV-YOLOv8: A small-object-detection model based on improved YOLOv8 for UAV aerial photography scenarios. *Sensors*, 23(16), p.7190.
12. Vijay, M.M., Kumar, O.P., Francis, S.A.J., Stalin, A.D. and Vincent, S., 2024. Enhancing high-speed digital systems: MVL circuit design with CNTFET and RRAM. *Journal of King Saud University-Computer and Information Sciences*, 36(4), p.102033.
13. Ranjit, K. C., Shrestha, B., & Ghimire, U. (2026). Optimizing CNNArchitectures for Steering Angle Prediction for Self-Driving Vehicles in Unstructured Roads: A Comparative Study of Activation Functions and Model Complexity. *ICCK Transactions on Machine Intelligence*, 2(2), 88–99. <https://doi.org/10.62762/TMI.2025.759110>
14. Kumar, A., & Lehal, G. S. (2026). Detection of Newspaper Layouts Using YOLO12. *ICCK Transactions on Machine Intelligence*, 2(2), 77–87. <https://doi.org/10.62762/TMI.2025.846033>
15. Wang,X., & Zhao,R.(2026). FedTLRec: FederatedRecommendation with Transformer-based Parameter Aggregation and LoRA Compression. *ICCK Transactions on Machine Intelligence*, 2(1), 65–76. <https://doi.org/10.62762/TMI.2025.882476>
16. Arya, S., Anju, & Onah, J. N. (2026). A Data-Driven Framework for Methane Emission Prediction Using Machine Learning Methods. *ICCK Transactions on Machine Intelligence*, 2(1), 53–64. <https://doi.org/10.62762/TMI.2025.782852>
17. Bai, Y., Hu, J., Tursun, E. & Yimit, H. (2026). Data-Driven Operational Assessment Method and Digital Twin System for Unmanned Surface Vehicles. *ICCK Transactions on Machine Intelligence*, 2(1), 38–52. <https://doi.org/10.62762/TMI.2025.444910>
18. Agarwal, S. K., & Gupta, Y. K. (2026). A Novel Approach of Progressive Transfer Learning for MRI Brain Tumor Classification Using VGG16 and MobileNet Architectures. *ICCK Transactions on Machine Intelligence*, 2(1), 28–37. <https://doi.org/10.62762/TMI.2025.367009>
19. Kumar, V., Saxena, A., & Chawla, L. (2026). Adaptive Learning Density Estimators for Tsallis Entropy and Kapur Entropy with Applications in System Training. *ICCK Transactions on Machine Intelligence*, 2(1), 12–27. <https://doi.org/10.62762/TMI.2025.317970>
20. Agarwal, S., Kandpal, M., Gupta, V. K., Aggarwal, S., & Jain, P. (2025). Secure and Decentralized Heart Sound Analysis using Federated Learning and Blockchain Technology. *ICCK Transactions on Machine Intelligence*, 2(1), 1–11. <https://doi.org/10.62762/TMI.2025.567350>
21. Pereira, M. S., de Azevedo, T. M., Ribeiro, C. T., Freire, A. I., Francisco, M. B., Pereira, J. L. J., & de Paiva, A. P. (2025). Discriminating Planted Capsicum Spp. Varieties via Machine Learning and Multivariate Data Reduction. *ICCK Transactions on Machine Intelligence*, 1(3), 166–185. <https://doi.org/10.62762/TMI.2025.385133>

22. Kumar, A. (2025). Clinical Text Analytics: Techniques, Deep Learning Models, and the Future of Medical Text Analytics. *ICCK Transactions on Machine Intelligence*, 1(3), 148–165. <https://doi.org/10.62762/TMI.2025.451731>
23. Malik, I. A., & Habib, G. (2025). AI Enabled Resource-Constrained Computing Architectures for IoT Devices. *ICCK Transactions on Machine Intelligence*, 1(3), 138–147. <https://doi.org/10.62762/TMI.2025.225921>
24. Bmotra, M., Kumar, R., & Dhiman, G. (2025). Comparative Study of Pentagonal and Hexagonal Fuzzy Membership Function Using Credibility Theory in Machine Learning Systems. *ICCK Transactions on Machine Intelligence*, 1(3), 127–137. <https://doi.org/10.62762/TMI.2025.922612>
25. Aggarwal, V., Aggarwal, S., Bhardwaj, A., & Gupta, V. K. (2025). Quantum Computing Essentials: Bridging Theory and Practice for New Learners. *ICCK Transactions on Machine Intelligence*, 1(3), 117–126. <https://doi.org/10.62762/TMI.2025.173543>
26. Sarvakar, K., & Rana, K. (2025). A Hybrid Framework Combining CNN, LSTM, and Transfer Learning for Emotion Recognition. *ICCK Transactions on Machine Intelligence*, 1(2), 103–116. <https://doi.org/10.62762/TMI.2025.572412>
27. Singh, S., Aggarwal, S., Singh, A., & Sharma, A. (2025). Integrating Artificial Intelligence and Machine Learning in Autism Detection via Gut Microbiome Analysis. *ICCK Transactions on Machine Intelligence*, 1(2), 90–102. <https://doi.org/10.62762/TMI.2025.682666>
28. Sharma, S., Devgan, K., Jangra, D., Bhardwaj, A., & Aggarwal, S. (2025). Neuro-Inspired Alert System for Air Quality Prediction Using Ensemble Preprocessing and SNN Classification. *ICCK Transactions on Machine Intelligence*, 1(2), 69–79. <https://doi.org/10.62762/TMI.2025.403059>
29. Khan, A., & Singh, J. (2025). A Novel Image Captioning Technique Using Deep Learning Methodology. *ICCK Transactions on Machine Intelligence*, 1(2), 52–68. <https://doi.org/10.62762/TMI.2025.886122>
30. Khan, M., Kumar, R., & Dhiman, G. (2025). A Hybrid Machine Learning Fuzzy Non-linear Regression Approach for Neutrosophic Fuzzy Set. *ICCK Transactions on Machine Intelligence*, 1(1), 42–51. <https://doi.org/10.62762/TMI.2025.561363>
31. Garg, M. (2025). Enhanced Reinforcement Learning-Based Resource Scheduling for Secure Blockchain Networks in IIoT. *ICCK Transactions on Machine Intelligence*, 1(1), 29–41. <https://doi.org/10.62762/TMI.2024.529242>
32. Gupta, S., Adhikari, U., Roy, D., & Hazra, S. (2025). IoT-Integrated Reinforcement Learning-Based Mine Detection System for Military and Humanitarian Applications. *ICCK Transactions on Machine Intelligence*, 1(1), 17–28. <https://doi.org/10.62762/TMI.2025.235880>
33. Myakala, P. K., Kamatala, S., & Bura, C. (2025). Privacy-Preserving Federated Learning for IoT Botnet Detection: A Federated Averaging Approach. *ICCK Transactions on Machine Intelligence*, 1(1), 6–16. <https://doi.org/10.62762/TMI.2025.796490>
34. Tiwari, S., Kumar, D., Mahajan, A., & Sachar, S. (2025). Emotion Detection from Speech Using CNN-BiLSTM with Feature Rich Audio Inputs. *ICCK Transactions on Machine*

Intelligence, 1(2), 80–89. <https://doi.org/10.62762/TMI.2025.306750>