HETRA: A Real-Time Traffic Detection and Directional Counting System Using YOLOv8 and DeepSORT

Kuldeep Kumar Shukla*, Ajai Kumar Singh
Civil Engineering Department, Motilal Nehru National Institute of Technology Allahabad, Prayagraj, 211004, India

***Abstract*—Rapid urbanization and economic growth in developing countries such as India have increased the demand for automated traffic analysis systems. However, heterogeneous traffic conditions present significant challenges to automated traffic analysis. This study proposes HETRA (HEterogeneous TRaffic Analysis), a real- time traffic monitoring framework that integrates YOLOv8 and DeepSORT for multi-class detection, tracking, and counting. For this, a custom traffic dataset was developed using real-world recorded traffic videos. The dataset consists of more than eight thousand annotated frames representing the fourteen traffic classes (motorized, non-motorized, and pedestrians). The proposed model leverages YOLOv8 for class identification and localization, while DeepSORT is used for multi-class tracking and directional traffic counting. The custom traffic dataset was used to train and assess the proposed HETRA model. The model achieved precision, recall, and mean average precision values of 71.15%, 65.51%, and 61.9%, respectively. The qualitative assessment of the model demonstrated its effectiveness under varying traffic scenarios and lighting conditions. Furthermore, it obtained an overall counting accuracy of 90%. These findings demonstrate the potential of the proposed framework for real-time traffic analysis and intelligent transportation systems.**

***Keywords*—Heterogeneous traffic; YOLOv8; DeepSORT traffic counting; intelligent transportation system**

I. INTRODUCTION

Over the last two decades, the fast-growing number of personal vehicles has resulted in an enormous increase in traffic congestion in cities, across developing countries, especially in India. Therefore, it is crucial to have an efficient traffic monitoring and management system for reducing travel time, decreasing fuel consumption, and supporting sustainable urban transportation goals. In recent years, Intelligent Transportation Systems (ITS) have gained tremendous attention as automated real-time traffic monitoring systems utilizing artificial intelligence (AI) and computer vision (CV) technology.

Vehicle detection, tracking, and counting are basic components of modern traffic monitoring systems. These are widely used for traffic flow analysis, congestion study, adaptive signal design, urban mobility planning, and road safety management. However, developing such an automated system for Indian traffic conditions remains highly challenging. The traffic patterns in India are fundamentally different from developed nations due to a heterogeneous traffic mix, - non-lane-disciplined behavior, poor roadway conditions, etc. The traffic in India consists of a vast composition of motorized and non-motorized vehicles competing for the same road space. Along with this, frequent overtaking, varying sizes, erratic lane practices, and dense traffic situations give rise to troublesome traffic class detection and tracking. Traditional intrusive sensor-based methods for traffic monitoring are inadequate as they provide only axle counts without traffic class differentiation. For traffic management, geometric design and other analyses, class-wise traffic counting is one of the mandatory requirements. These challenges accentuate the immediate need for reliable automated traffic analysis techniques capable of generating accurate traffic information, including vehicle classification, class-wise traffic volume, and movement patterns. Such information serves as the foundation of ITS, enabling applications including adaptive signal management, congestion prevention, accident detection, safety assessment, and also long-term transportation planning [1].

Conventional approaches used for traffic monitoring are: manual traffic counts, loop detectors, pneumatic tubes, radar and infrared sensors, and conventional image processing-based techniques. These methods exhibit considerable limitations in scalability, high installation cost, poor adaptability, and reduced accuracy under dynamic traffic circumstances [2]. The image processing approaches for traffic detection are based on traditional CV techniques like background subtraction [3], Support Vector Machine (SVM) [4], etc. However, their robustness is often compromised by lighting conditions, shadows, and occlusions encountered in real-world scenarios.

In recent years, deep learning-based computer vision techniques have demonstrated remarkable success in identifying, tracking, and traffic analysis [5]. Among these developments, convolutional neural networks (CNNs) have become the foundation of modern object detection algorithms. Existing deep learning-based detectors are broadly categorized into two-stage and one-stage approaches. Two- stage methods, including R-CNN [6], Fast R-CNN [7], and Faster R-CNN [8], first generate candidate object regions before performing classification, which generally results in high detection accuracy. However, their high latency in inference as well as computational complexities are major drawbacks in real-time traffic monitoring and surveillance applications. To address these constraints, single-stage detectors like Single Shot Detector (SSD) [9] and the You Only Look Once (YOLO) [10] family are introduced. SSD utilizes multi-scale feature maps to identify objects of diverse dimensions, yet it often has

*Corresponding author.

difficulties in detecting tiny objects, which is essential in traffic situations involving bicycles, bikes, and pedestrians [9].

Among these techniques, the YOLO method redefined object detection as a single regression task, allowing real-time identification with competitive detection precision [10]. YOLO9000 demonstrated the viability of integrated identification and categorization across several object categories while preserving real-time speed [11]. In YOLOv3, multi-scale predictions and residual learning were incorporated, which greatly promoted the accuracy of object detection of small objects [12]. YOLOv4 further boosted detection capabilities with optimized architecture and advanced feature fusion techniques, and reached state-of-the-art performance on test sets with high processing speeds [13]. These upgrades brought YOLO-based detectors into the spotlight as a leading tool for real-time object detection in smart traffic surveillance systems. However, they also showed their limitations in dense as well as mixed traffic conditions.

Recent research studies have investigated the use of YOLOv5 and YOLOv7 for urban traffic monitoring. Farid et al. established a real-time vehicle identification system using YOLOv5 and validated its efficacy in traffic surveillance contexts [14]. Likewise, traffic counting models created on deep learning-based techniques mostly emphasized organized traffic scenarios with a restricted range of vehicle classes [15].

Most of the existing studies primarily address structured traffic scenarios with few conventional vehicle classes. Although YOLOv8 and DeepSORT-based models have previously been applied for traffic monitoring, their assessment under heterogeneous traffic conditions remains insufficiently investigated. To address this limitation, this study develops and evaluates HETRA (HEterogeneous Traffic Analysis), an integrated YOLOv8-DeepSORT framework, for real-time detection, tracking, and class-wise directional counting of fourteen traffic categories under Indian traffic conditions. For this, the proposed framework was trained on a prepared custom traffic dataset of 8,476 manually annotated frames representing fourteen different traffic classes related to motorized vehicles, non-motorized vehicles, and pedestrians. The dataset includes region-specific traffic participants like auto, e-rickshaw, tempo, and horse-cart, which are often neglected in publicly available traffic datasets. The effectiveness of the developed model was evaluated using conventional metrics such as precision, recall, mean Average Precision (mAP), F1-score, confusion matrix, and directional counting accuracy. The novelty of the study is the thorough assessment of an integrated multi-class traffic analysis framework in dense, non-lane-based scenarios with frequent occlusions and substantial variations in traffic classes and size.

The major contributions of the study are as follows:

- Development of a custom traffic dataset consisting of 8476 manually annotated image frames representing fourteen different traffic classes in Indian traffic environments.
- Development and evaluation of HETRA- a YOLOv8 and DeepSORT-based framework for real-time multi-class detection, tracking, and directional traffic counting using the custom dataset.
- Assessment of HETRA under varying traffic densities and lighting conditions using quantitative and qualitative performance analysis.
- Demonstrate the suitability of the anticipated approach for real-time traffic monitoring, directional class-wise traffic flow estimation, and ITS applications.

II. Related Work

YOLOv8 brought significant architectural enhancements like the anchor-free detection strategy, decoupled detection heads, and refined loss functions. A number of recent works have used YOLOv8 in traffic-related tasks. Enhanced YOLOv8 has been utilized for identifying moving objects in videos, which showed superior accuracy and convergence characteristics compared to earlier YOLO variants [16]. Survey studies have recognized YOLOv8 as a viable model for real-time detection applications; however, its comprehensive evaluation under complicated, unstructured traffic situations is still restricted [17-19]. The YOLOv8 models suggested for vehicle counting in road traffic indicate accurate object detection performance in controlled environments [20].

Subsequent enhancements in YOLOv8 continued improvements in model performance under complex environments. Different architectural modifications in YOLOv8 were employed for the estimation of traffic flow parameters to improve the model precision [21]. Various works have utilized YOLOv8 for improving traffic analysis in ITS applications but only for limited classes of vehicles [22]. The real-time detection of small and distant traffic objects has remained one of the crucial challenges in traffic system monitoring. Many researchers employed different variants of improved YOLOv8 models for detecting small objects. Lightweight and modified YOLOv8 models were reported for the detection of small traffic signs and pedestrian-vehicle detection, with better precision in difficult environmental conditions [23, 24]. UAV-based studies also explored several improvements over YOLOv8 to detect small vehicles from an overhead view, which is quite different from the roadside camera viewpoint [25, 26]. While these works attained performance gains, most dealt with particular object categories or aerial imagery and didn't look into the complex issues of ground-level mixed traffic with overlapping objects and diverse vehicle types.

Multi-object tracking (MOT) has emerged as one of the critical aspects of traffic monitoring systems. DeepSORT (Simple Online and Realtime Tracking with Deep Association Metric) is one of the most popular approaches to MOT, which is achieved by integrating Kalman filtering with deep appearance features to maintain stable traffic class IDs between successive frames [27]. A vision- based vehicle detection and counting using YOLOv3 and Oriented FAST and Rotated BRIEF (ORB) tracking demonstrated effective performance under highway traffic scenarios [28]. Durve et al. performed a comprehensive benchmark analysis of YOLOv5 and YOLOv7 in combination with DeepSORT and highlighted the correlation between tracking precision and detector efficiency. It also emphasized the efficiency of the detector in congested environments [29].

Recent research studies have integrated YOLOv8 with MOT algorithms like DeepSORT to overcome the constraints of detection-only systems. The authors have employed a YOLOv8-DeepSORT-based method for estimating the traffic flow parameters with improved recognition consistency between consecutive frames [30]. On the other hand, some vehicle detection and tracking systems using YOLOv8 were effective in a controlled surveillance environment [31]. An integrated YOLOv8/v9 model with the ByteTrack and CLAHE (image enhancement technique) was used to evaluate the counting and tracking performance under varying environmental and traffic situations. The results demonstrated effective improvement in model performance, especially under poor lighting conditions [32].

The availability of image datasets of various traffic classes also continues to be a significant barrier in diverse traffic research. Early studies concentrated mostly on organized highway situations and lane-specific traffic scenarios. A range of benchmark datasets, including PASCAL VOC [33], MS COCO [34], KITTI [35], Cityscapes [36], and Open Images [37], has enabled researchers to attain substantial gains in detection accuracy. Nonetheless, these datasets mostly reflect traffic circumstances in developed nations and may not adequately depict the intricacies of diverse traffic scenarios often encountered in developing countries. Paranjape and Naik presented the DATS_2022 dataset, which highlights 45 classes including animal, board, poles, and 12 vehicle classes, but its applicability for traffic analysis is not reported [38]. This emphasizes that there is a need for a realistic dataset for unstructured traffic conditions with appropriate size, class variety, and practical applicability.

The literature studies indicate that most of the YOLOv8-based traffic monitoring works focused on general and structured traffic datasets, failing to deal with the complexities encountered in developing countries [15, 21]. These studies mostly concentrated on very few motorized traffic classes, while ignoring traffic categories like pedestrians, bicycles, and other informal modes like rickshaws, animal-driven carts, etc. [20, 22]. Very limited research on real-time multi-class vehicle detection and tracking is reported for Indian traffic conditions. The comprehensive fusion of tracking of the involved traffic objects and their class-wise vehicle counting was missing, which restricted their use in overall traffic analysis [16, 31]. Architectural developments of YOLOv8 have increasingly attracted more attention over time, but limited research has been conducted to evaluate its reliability under real-world traffic environments. Therefore, the use of models is largely confined to region-specific datasets [21, 22, 30].

III. Methodology

This research primarily offers an inclusive vision-based paradigm for the real-time traffic detection and counting system under heterogeneous traffic conditions. The workflow of the developed methodology is depicted in Fig. 1, illustrating the chronological phases from custom traffic dataset preparation to performance evaluation of the HETRA framework. The primary phases of this approach are creating a custom traffic dataset, annotating all instances, training the YOLOv8 model, evaluating the model performance, and integrating with DeepSORT for tracking multiple traffic classes and their directional counting across consecutive frames.

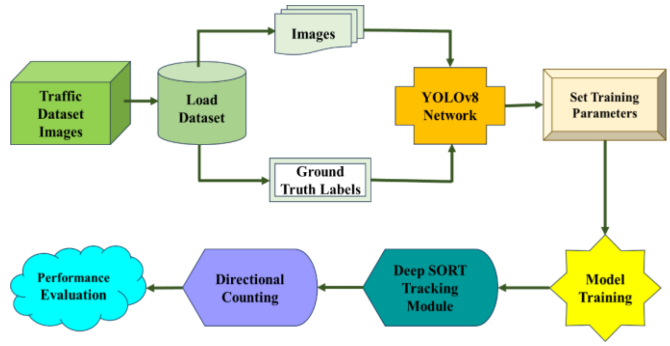

Fig. 1. Overall architectural workflow of proposed HETRA framework.

A. Dataset Preparation

The dataset was developed using actual traffic videos obtained from various urban and rural roadways in India. The dataset is intentionally enriched with various perspectives such as lighting conditions, traffic densities, background intricacy, etc. The dataset comprises fourteen traffic categories- namely auto, bike, bicycle, bus, car, horse-cart, e-rickshaw, pedestrian, pickup, rickshaw, tempo, tractor, truck, and van reflecting the diverse and complex composition of Indian road traffic.

The video footages of were chosen from different traffic conditions at regular intervals to prevent redundancy. It was furnished by removing low-quality or inappropriate image frames after visual inspection. Thereafter, high-quality image data was processed to the labelling stage. This Data was augmented by changing the original traffic images through modifications such as flipping and cropping before annotation to avoid overfitting and improve model flexibility.

Python-based 'LabelImg' annotation tool was utilized to label superior image frames manually in YOLO format (.txt). All annotations were manually inspected to verify the correctness of class labels and bounding-box localization. Incorrect or inconsistent annotations were corrected before finalizing the dataset for model training. A sample of image frames is shown in Fig. 2.

Fig. 2. Sample illustration of the annotation process.

Extra care was taken in labelling the bounding boxes of the concealed, overlapping, and smaller objects, which are more effective in practical scenarios for detection. Meticulous attention was paid while drawing the bounding boxes around the traffic classes. Each label file contained the class index (numeric identifier referring to fourteen traffic categories) and box coordinates (‘center_x’, ‘center_y’, width and height) in standardized format.

A custom traffic dataset of 8476 frames consisting of 22886 instances was thus created, which corresponds to an average of 2.7 instances per frame. The diversified instance distribution of different traffic classes was analyzed to find the less represented traffic class, as presented in Fig. 3. No artificial data was generated to keep realistic traffic characteristics, while some horizontal image flipping was applied to minor categories. The common traffic classes such as cars, bikes, trucks, pedestrians, and pickups have a significantly higher number of instances compared to the less frequent classes like van, horse-cart, and rickshaw. The class ‘Car’ consists of the highest number of instances (6091), whereas less common classes like ‘Horse-cart’ and ‘Rickshaw’ consist of 127 and 132 instances, respectively. This exhibits an inherent class imbalance ratio of around 48:1, which is realistic for traffic conditions encountered on Indian roads. The class-wise performance metrics were reported to evaluate the influence of class imbalance. To enhance model generalization and mitigate overfitting, the dataset was partitioned into training, validation, and testing subsets.

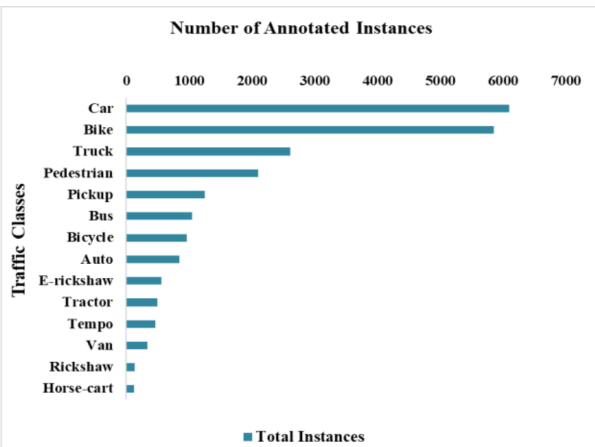

Fig. 3. Instance distribution of annotated traffic classes used in the custom dataset.

B. YOLOv8 Model Architecture

Fig. 4 shows the simplified YOLOv8 network used in this study. Improved feature extraction, anchor-free detection, and enhanced localization accuracy with faster inference performance are the main characteristics of YOLOv8, which makes it suitable for implementation under heterogeneous traffic conditions. Scalable pre-trained models from YOLOv8 can be fine-tuned on custom datasets, and this versatility allows strong learning of multiple traffic classes and complicated visual patterns (visually similar traffic classes, overlapping etc.).

The YOLOv8 architecture consists of three basic components: the backbone, neck, and detection head network. Each of them contributes to overall detection performance. The backbone network consists of Cross Stage Partial networks (CSP-net) based on two convolution (C2f) modules along with a Spatial Pyramid Pooling Fast (SPPF) block [39]. It extracts intricate low- as well as high-level semantic characteristics from input picture frames through a series of convolutional operations and feature fusion layers. This ensures better extraction of feature representation for small and partially occluded classes.

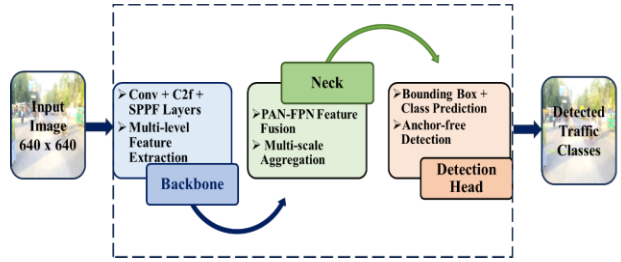

Fig. 4. YOLOv8 architecture illustrating the backbone, neck, and detection head for multi-class traffic detection.

The neck network employs a Path Aggregation Network combined with a Feature Pyramid Network (PAN-FPN) to integrate feature representations extracted at multiple scales. It is an intermediate processing part positioned between the backbone and the detection head. It is responsible for the aggregation of feature maps produced at the different backbone stages. This multi-scale feature integration facilitates efficient detection of traffic classes at varying distances and scales within the frame, thereby improving detection reliability in congested traffic conditions.

At the end, an anchor-free detection head is utilized that simultaneously predicts the bounding box coordinates, confidence scores, and class probabilities. It makes the model more robust to scale and aspect ratios. The anchor-free approach diminishes the computational intricacy and enhances the detection performance for irregular-shaped and small objects. This improves the overall efficiency of the model under dense and mixed traffic movements.

C. Multi-Class Tracking and Counting

The YOLOv8 model detects multi-class vehicles efficiently, but their identity preservation across successive video frame poses still an issue as traffic keeps changing. Therefore, the trained detector model is integrated with the DeepSORT algorithm to enable continuous multi-class tracking. Based on detection results, the DeepSORT technique provides a distinct identity to each tracked class. A Kalman filter is utilized for predicting future locations based on the previous position and movement of each traffic class track over various video frames. New detections are used to update and correct these predictions. Linking detections to existing class tracks is done using the Hungarian algorithm and deep appearance matching [27].

The hyperparameters within DeepSORT govern the timing and trajectory prediction of the traffic participants. Some of the tracking parameters are tuned specifically for congested traffic scenarios. The hyperparameter matrix used in the configuration file is summarized in Table I. Non-Maximum Suppression (NMS) was utilized to remove duplicate overlapping detections and preserve the bounding box with the highest level of confidence for each object. In addition to this, Intersection-over-Union (IoU) was applied in the process of data association while applying DeepSORT to correlate new detections with old tracks based on temporal overlap. Furthermore, frequent occlusions of small-sized vehicle classes (auto, bike, bicycle, etc.) while moving adjacent to and behind the large vehicle class (bus, truck etc.) are noticed especially in mixed traffic conditions. Therefore, the MAX_AGE hyperparameter becomes an important aspect, especially in traffic monitoring applications. In this study, MAX_AGE was empirically set to 70 frames based on preliminary investigation and guidance from previous DeepSORT based tracking studies. This value provided a practical balance between the identities over short occlusions and prevented retention of prolonged tracking. This reduced the risk of double counting, while limiting the surplus computational overhead.

TABLE I. DEEPSORT TRACKING PARAMETERS

Hyperparameters	Operational Value	Function in Traffic Environments
MAX_DIST	0.2	Establish upper limit for cosine distance for visual appearance matching, helps to maintain correct traffic class identities
MIN_CONFIDENCE	0.3	Deny all class detections having a confidence score below 30%, reducing false positive detections
NMS_MAX_OVERLAP	0.5	Non-Maximum Suppression (NMS) level to eliminate duplicate bounding boxes for a single class with more than 50%
MAX_IOU_DISTANCE	0.7	Maximum Intersection over Union (IoU) distance allowed for overlapping bounding boxes
MAX_AGE	70	Retains class track identity up to 70 successive frames even in partial obstruction and missed detection
N_INIT	3	At least 3 successive stable detection frames required before allocating a tracking ID to reduce false track ID
NN_BUDGET	100	Stores 100 most recent frames of the class appearance feature

During the tracking phase, each detected traffic class is allocated a unique identity number which is maintained over successive video frames. Due to this consistent identification, it is possible to keep track of each traffic category in moving scenes and ensure non-repetitiveness of identity for the same object. This ensures accurate computation of traffic volume count, while minimizing the duplicity in detection and identity conflict issues.

Traffic counting is implemented through the use of a virtual counting line (VCL), passing through a defined point within the video frame. This line acts as a reference for counting the traffic. A trajectory buffer with a deque length of 20 was used for each tracked class to facilitate directional counting. The stored centroid history was utilized to analyse the movement and direction with respect to VCL. The bottom center of bounding box of each class continuously observed and counted traffic class is generated as it crosses the VCL. This minimizes the duplicate counts caused by repeated detections in consecutive frames. Traffic categories crossing the VCL while moving in the downward direction in a frame are reported as 'Number of Vehicles Entering,' and those crossing the VCL while moving in the upward direction are reported as 'Number of Vehicles Leaving'. Class-wise counters are established for each direction separately. This facilitates precise class-wise estimation of directional traffic count in real-world traffic scenarios.

D. Training Strategy and Loss Functions

In this study, YOLOv8 version 'l', was chosen as the traffic detection model because it offers a practical balance between detection performance and computational overhead. The model was fine-tuned using the custom traffic dataset consisting of a total of 8476 manually annotated image frames representing fourteen different traffic participants. The dataset was divided into training, validation, and testing subsets in the ratio of 66%, 14%, and 20%, respectively. Transfer learning is employed by initializing pretrained YOLOv8l weights with an input image resolution of 640 x 640 during training. The model was trained and optimized with loss components such as box, classification, and Distribution Focal Loss (DFL) over 200 epochs. The weights corresponding to the best performance were selected for subsequent evaluations. The major training parameters along with implementation aspects are summarized in Table II.

TABLE II. SUMMARY OF TRAINING SETUP

Parameter	Description
Model	YOLOv8l
Dataset Size	8476 images
Number of Classes	14
Input Image Size	640 × 640
Training Epochs	200
Batch Size	16
Optimizer	Stochastic Gradient Descent (SGD)
Learning Rate	0.01
Platform	Google Colab
GPU	NVIDIA A100 SXM4-40GB
Loss Function	Composite Loss (CIoU + BCE + DFL)

E. Loss Functions

The composite loss function was used to train the model. This minimizes the loss for localization accuracy and classification performance, and enhances the bounding box precision. Eq. (1) shows the total loss function (L_{TOTAL}) [12, 40, 41].

$$L_{TOTAL} = \lambda_{BOX} L_{BOX} + \lambda_{CLS} L_{CLS} + \lambda_{DFL} L_{DFL} \quad (1)$$

where, L_{BOX}, L_{CLS}, and L_{DFL} symbolize the loss for bounding box regression, classification, and distribution focal loss, respectively. The λ_{BOX}, λ_{CLS}, and λ_{DFL} are corresponding loss weightage coefficients.

To enhance localization precision, bounding box regression loss employs the Complete Intersection over Union (CIoU) metric. This loss accounts for overlapping area, center distance, and the aspect ratio uniformity between ground truth and predictions as shown in Eq. (2) [40].

$$L_{BOX} = L_{CIoU} = 1 - IoU + \frac{\rho^2(b,b^{gt})}{d^2} + \beta \upsilon \quad (2)$$

where, ρ (b, b^{gt}) represents the Euclidean distance between the centers of the predicted and the ground-truth bounding box, 'd' is the diagonal length of the smallest enclosing box, and 'υ' is aspect ratio uniformity and 'β' is a weighting parameter that controls the influence of the aspect ratio term in the CIoU loss function.

The classification loss 'L_{CLS}' is computed using Binary Cross-Entropy (BCE) loss as expressed in Eq. (3)[12]to penalize incorrect class predictions.

$$L_{CLS} = L_{BCE} = -\sum_{i=1}^{C=14}[y_i \, log(\hat{y_i}) + (1 - y_i)log(1 - \hat{y_i})] \quad (3)$$

Where C = 14 is the total number of traffic classes, y_i represents the ground truth class label, and $\hat{y_i}$ predicted probability for the i^{th} class. This loss function accounts for the error between predicted and actual labels, thereby making it well-suited for multi-class detection problems.

DFL is utilized to refine bounding box localization and improve detection accuracy. This is especially important for small and partially occluded traffic objects prevalent in heterogeneous traffic. The mathematical formulation of DFL is given in Eq. (4) [41].

$$L_{DFL} = \sum_{k=1}^{4}\sum_{i=0}^{n}[p_i \log(\hat{p_i})] \quad (4)$$

where, pi and $\hat{p_i}$ are the actual and predicted discrete probability distributions for each bounding box coordinate.

The integration of these loss functions strengthens the detector model in detecting and localizing the classes of different shapes and sizes.

F. Evaluation Metrics

The performance of the detector model was assessed by using standard evaluation metrics. These are: Precision, Recall, F1-score, as well as mean Average Precision (mAP). These parameters were calculated using the confusion matrix elements, including True Positives (TP), False Positives (FP), True Negatives (TN), and False Negatives (FN), to assess the effectiveness of the model.

Precision represents the ratio of accurately detected to the total number of detected objects, which is defined as:

$$Precision = \frac{TP}{TP+FP} \quad (5)$$

Recall measures the ratio of correctly detected to the total number of ground truth objects and is given by:

$$Recall = \frac{TP}{TP+FN} \quad (6)$$

The F1-score offers a fair assessment of accuracy and recall that is calculated as

$$F1\_Score = \frac{2*Precision*Recall}{Precision+Recall} \quad (7)$$

The area under the Precision-Recall (P-R) curve represents Average Precision (AP) for the given class. The P-R curve illustrates the balance that exists between detection precision and comprehensiveness at different confidence levels. It is calculated by integrating precision with respect to recall over the interval [0,1].

$$AP = \int_0^1 Precision(Recall)d(Recall) \quad (8)$$

The mAP is derived by averaging AP over all 'N' traffic classes (here N=14).

$$mAP = \frac{1}{(N=14)}\sum_{i=1}^{N=14} AP_i \quad (9)$$

The mAP@0.5 refers to mean Average Precision evaluated at a specific threshold IoU value of 0.5, whereas mAP@0.5:0.95 is a metric that averages the mAP values for various IoU thresholds ranging from 0.5 to 0.95 at intervals of 0.05. It provides a rigorous and thorough assessment for testing the accuracy of object detection and localization at various thresholds.

IV. Results and Discussion

This section presents a comprehensive evaluation and analysis of the HETRA framework for real-time detection, tracking, and directional counting of multiple-class traffic. The effectiveness of the trained detector was assessed using universal evaluation metrics such as precision, recall, mean Average Precision (mAP), F1 measure, as well as analysis of the confusion matrix. Additionally, to assess robustness and real-world applicability, the proposed system was analysed using multiple recorded traffic videos with different levels of traffic density and lighting conditions.

A. Training Performance Analysis

Fig. 5 illustrates the comprehensive performance of the detector model over 200 epochs. The graph shows progressive reduction in box loss, classification loss, and DFL throughout the training process.

The training curves indicate stable convergence, with no significant indications of instability. Consequently, model generalization was assessed through validation metrics, such as precision, recall, and mean Average Precision (mAP), which is consistent with recent YOLOv8-based object detection studies. It may be noted that the model was validated throughout the training process, but the val/cls_loss recorded by YOLOv8 remained zero and therefore did not produce a visible curve in Fig. 5. As the training epoch increases, a steady improvement in precision and mAP values was observed, while moderate fluctuation was noticed in recall. This demonstrates efficient feature learning and generalization capability of the detector model.

B. Detection and Classification Performance Evaluation

The precision–confidence, recall–confidence, and F1-confidence curves obtained during model evaluation are illustrated in Fig. 6. Gradual improvement in the precision curve at higher confidence values was observed due to a reduction in false positives. On the other hand, the recall value decreased at a confidence threshold due to some true values being missed during confined detection. The F1-confidence curve achieved a peak F1-score of 0.63 at a confidence threshold of 0.33. This shows an optimal balance between precision and recall at a moderate threshold.

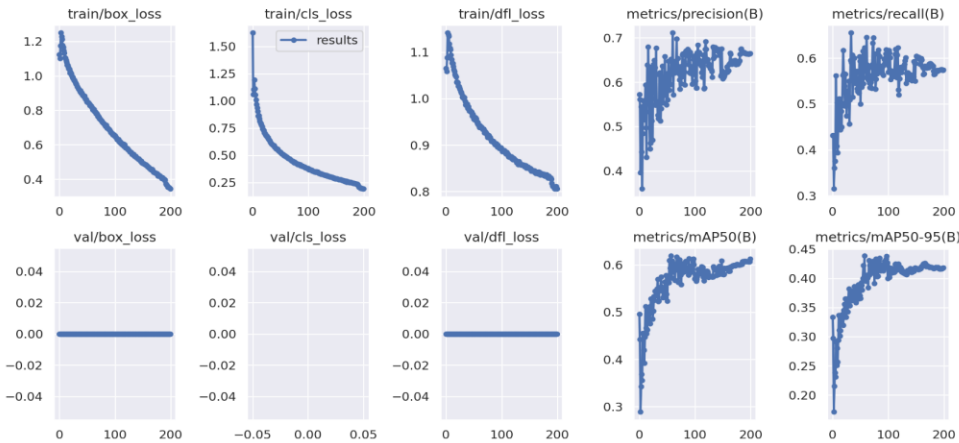

Fig. 5. Comprehensive training performance assessment of the detector model.

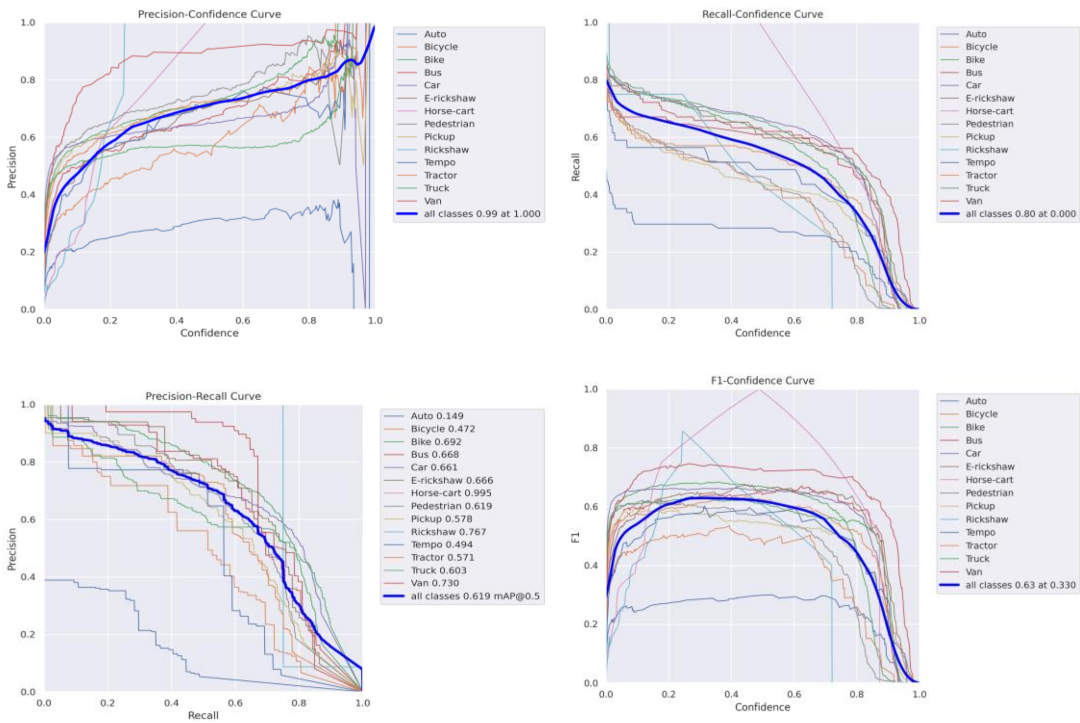

Fig. 6. Precision-Confidence, Recall-Confidence, Precision-Recall, and F1-Confidence curves of the YOLOv8l model for 14 different traffic classes.

The precision-recall curve indicated stable detection performance for major and visually distinguishable classes such as cars, buses, pickups, and trucks. Comparatively lower precision and recall values were observed for less- frequent classes including horse-cart and rickshaw, primarily due to the limited number of training instances available for these categories. Although the auto and bicycle classes contained a substantial number of instances, their detection performance also remained comparatively lower. This suggests that factors beyond class imbalance, including smaller object sizes, repeated occlusions, and high intra-class variability, significantly affected the detection performance under heterogeneous traffic conditions.

The trained model achieved precision, recall, mAP@0.5, and mAP@0.5:0.95 values of 71.15%, 65.52%, 61.9%, and 43.85%, respectively. These results indicate efficient detection and localization capability for various traffic participants. Whereas a comparatively lower performance was noticed for certain class may be due to less representation, small size, along with partial occlusions under dense traffic situations. The complete detection performance of the model can be attributed to multi-scale feature extraction and the diversity of the custom traffic dataset. This enabled the model to learn more vigorous features even in varying traffic densities, weak lane discipline, and frequent occlusions.

Although the performance curves provided a comprehensive assessment, the normalized confusion matrix offers detailed insights into classification behavior across different traffic classes, as presented in Fig. 7. There was considerable diagonal dominance for the most classes of traffic, indicating reliable multi-class detection and classification. Classes such as bike, bus, car, horse-cart, e-rickshaw, and truck achieved higher classification accuracy due to their visuals and adequate representation within the traffic dataset. Comparatively lower accuracy was observed for classes like auto and rickshaw, which may be due to their visual similarity and small scale. Some discrepancies were also observed between visually similar categories, especially rickshaw vs. e-rickshaw as well as van vs. car. In addition, moderate confusion with the background was also noticed for small and partially hidden traffic participants. Despite these challenges, the confusion matrix illustrates the robustness and effectiveness of the model for detection and localization of a wide range of traffic participants.

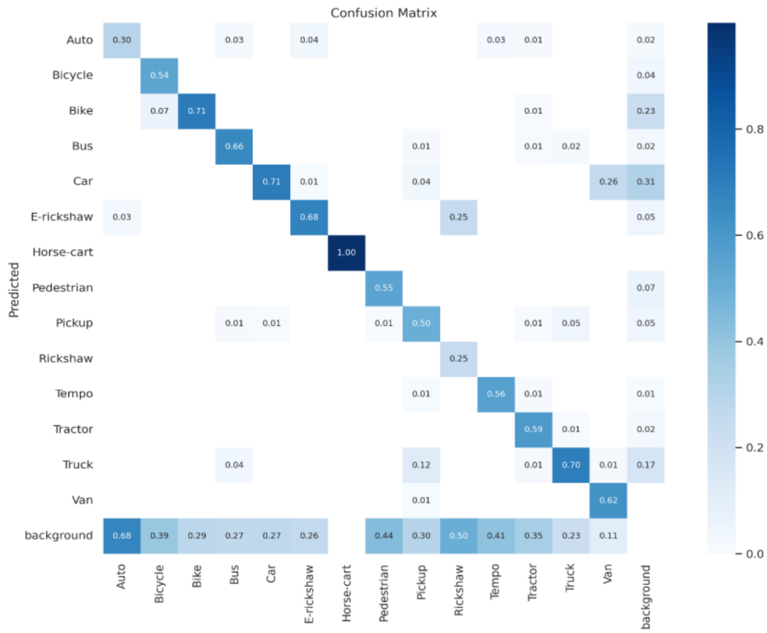

Fig. 7. Normalized confusion matrix obtained for custom dataset.

Fig. 8. Detection, localization and multi-class tracking results of the HETRA model under real-world traffic conditions.

C. Qualitative Analysis

The real-world traffic videos, each of approximately 5 minutes' duration, were recorded at the Teliyarganj Intersection in Prayagraj city and were then analyzed using the developed HETRA model. Fig. 8 illustrates a few sample frames representing the qualitative performance of the model under different lighting and traffic conditions. The model was evaluated under normal weather conditions. The visual outcomes indicate that multiple traffic classes were detected and localized simultaneously for large vehicles like buses and trucks, as well as smaller road users like bicycles and pedestrians.

The visual outcomes also show a substantial discrimination between visually similar traffic classes such as 'bicycles & bikes' and 'car & van'. This results in fewer identity swaps among various traffic categories. Furthermore, the model maintained reliable detection performance across multiple traffic classes at various distances and scales within the surveillance frames.

The successive sample frames also indicate the tracking results of various traffic participants. These demonstrate consistent preservation of class ID even under continuous moving and overlapping traffic flow. As shown in Fig. 8, the tracking sequences of car (IDs 105 & 126) crossed each other with their unique IDs. Also, small-class traffic like a bike (ID 127), moving in proximity to the car, preserved its consistent trajectory in the successive frames. It is also observed that the vulnerable road user (pedestrian ID 1116) retained its identity while crossing the road near the car (IDs 961&1122). These findings validate the suitability of the HETRA model for its sustained tracking capability under various traffic participants (from public transit mode bus to low mobility modes such as bicycles, e-rickshaws, as well as pedestrians) in mixed traffic conditions.

Some intermittent miss-detections and slight localization lapses were also observed by the model for small traffic classes under occluded conditions. However, overall qualitative performance of the model is found to be adequate for real-time traffic monitoring and applications. The visual outcomes validate the efficacy and practical utility of the DeepSORT-based tracker for multiclass traffic analysis under heterogeneous traffic conditions.

The counting performance was evaluated using multiple representative traffic videos. Unlike the conventional method, this study focuses on accomplishing bi-directional traffic flow, while maintaining separate real-time records for entering and leaving traffic movements. The class-wise directional traffic counting relies on the unique class ID, trajectory analysis, and a virtual counting line that works as a reference boundary within video frames. Representative example frames of class-wise directional traffic counting results are shown in Fig. 9.

As evident from Fig. 9, the traffic class is counted only once, and it is done when the traced traffic trajectory crosses the virtual counting line in the specific direction. This eliminates the duplicate counting of vehicles across successive frames and yields the desired class-wise traffic counting. This track-assisted counting approach facilitates precise estimation of directional traffic volume. The results also indicate that the developed approach supports the classification and counting of region-specific classes like auto and tempos alongside standard traffic classes.

The counting statistics were validated using multiple real-world traffic video recordings. Manual counts for both entering and leaving directions were acquired by visual inspection and thereafter compared with the automated counts obtained by HETRA. The traffic categories, namely horse-cart, tractor, and rickshaw, were not observed in the selected videos due to their absence in the recorded videos. Therefore, the counting results for these classes could not be evaluated. The directional traffic counting results corresponding to representative traffic phases are presented in Table III. The directional counting accuracy ranged from 80% to 96% across different traffic classes. An aggregate 1320 traffic classes were manually counted in cross-sample videos, whereas a total of 1181 participants estimated by HETRA achieved an overall counting accuracy of approximately 90%.

TABLE III. CLASS-WISE DIRECTIONAL TRAFFIC COUNTING PERFORMANCE OF THE MODEL

Traffic Class	Entry Count		Exit Count	
	Model	Manual	Model	Manual
Auto	72	84	83	96
Bicycle	39	46	54	66
Bike	141	152	167	179
Bus	45	50	34	38
Car	127	132	105	113
E-rickshaw	27	32	41	47
Pedestrian	33	37	24	27
Pickup	28	33	34	42
Tempo	40	47	36	41
Truck	7	8	10	11
Van	19	21	15	18
Total	**578**	**642**	**603**	**678**

Most recent studies primarily reported the detection accuracy and counting performance, whereas their computational efficiency was mostly ignored. The inference performance of the developed model was tested on an average compute in Google Colab using an NVIDIA T4 GPU with 16 GB memory. The mean preprocessing, inference, and post-processing times were 0.5 ms, 20.2 ms, and 2.2 ms per frame, respectively. This results in an overall processing time of approximately 23ms per frame, nearly 44 FPS (frames per second). Since the evaluation videos were recorded at 30FPS, the achieved processing speed is higher than the recording rate. Although GPU memory utilization was not explicitly profiled, the observed inference performance demonstrates the computational efficiency and practical scalability of the HETRA framework for real-time heterogeneous traffic analysis.

Fig. 9. Directionally classified traffic volume by the HETRA model in real time under heterogeneous traffic conditions.

Minor counting errors were noticed in rare circumstances, including highly congested conditions, abrupt direction changes, and intermittent ID changes, near the counting line. However, these remain limited and had a minute effect on the overall traffic counting results. While a slight reduction in accuracy was observed in relatively low light conditions, the overall results demonstrate the effectiveness of the model under different lighting conditions. Less frequent traffic categories like horse-carts, rickshaws, and tractors were not observed during representative counting assessment and so could not be included in the analysis.

Direct comparison of the reported performances should be interpreted with caution because of variation in dataset, class compositions, environmental conditions and evaluation methodologies. A concise comparative analysis with selected relevant studies is presented in Table IV. It can be observed from Table IV that most of the existing studies reported higher detection and counting performance in relatively structured traffic streams involving three or four vehicle classes only. In contrast, the HETRA system was tested under more challenging heterogeneous traffic conditions comprising fourteen traffic categories, including region-specific classes such as auto, tempo, horse-cart, and rickshaw. Therefore, the lower mAP@0.5 achieved by the HETRA framework is mainly due to the complexity associated with multi-class detection under dense, frequent occlusions and weak-lane discipline and class imbalance issues. Instead of optimizing for better performance for a limited number of traffic categories, the proposed framework was designed to offer robust multi-class detection, tracking, and directional counting under varying illumination levels, frequent occlusions, and weak lane discipline.

TABLE IV. COMPARATIVE ANALYSIS WITH EXISTING TRAFFIC MONITORING STUDIES

Study	Dataset	No. of classes	Precision (%)	Recall (%)	mAp@0.5 (%)	Counting Accuracy (%)
Duan et.al.[20]	VisDrone2019	10	60.29	46.34	49.64	-
Song et.al.[28]	Highway Scene	3	88.0	89.0	87.88	93.2
Sathiyaprasadet.al.[30]	Urban	4	96.8	96.5	96.3	90
Akoushideh et.al.[32]	Extracted from MSCOCO	3	98.0	97.0	98.0	88.84
Proposed HETRA framework	**Custom**	**14**	**71.15**	**65.51**	**61.9**	**90**

Overall, the results of the HETRA model indicate its capability for multi-class detection, tracking, and class-wise directional counting with commendable accuracy under both normal and relatively low light conditions. This emphasizes the model's potential for real-time traffic monitoring, class-wise traffic movement analysis, and other ITS applications.

V. CONCLUSION

The proposed HETRA, integrating YOLOv8 and DeepSORT, was developed and evaluated for real-time detection, tracking, and directional counting under heterogeneous Indian traffic circumstances. The developed system effectively handled mixed traffic streams consisting of motorized traffic classes, non-motorized traffic classes, and pedestrians by addressing the complexity of urban traffic in developing countries.

A custom annotated traffic dataset consisting of fourteen classes was created to train and evaluate the model under typical urban conditions, including congestion, occlusions, and unstructured traffic movement. The experimental results confirmed that the suggested approach can reliably detect, track, and directionally count the traffic participants, making it suitable for real-time traffic monitoring. The findings further demonstrated the model's robustness for congested traffic, poor lane discipline, and mixed traffic under varying light conditions. Importantly, the detection and counting of slow-moving and vulnerable road-user categories like bicycles and pedestrians alongside motorized traffic highlight its potential for comprehensive traffic surveillance.

Apart from its detection and counting performance, the proposed framework offers a solid foundation for the development of an AI-enabled intelligent transportation system. The classified traffic volume information, generated by the proposed methodology, can help with modal share analysis, congestion monitoring, road safety assessment, and adaptive traffic signal control. Consequently, the system offers a scalable and cost-effective solution for real-time traffic monitoring and future smart city traffic management, especially in developing countries.

Despite the promising performance, the developed system experiences some challenges under severe occlusions and low light conditions. Comparatively lower performance was noticed for some less-frequent traffic classes, which may be attributed to class imbalance. The current study is limited to normal weather conditions, but the performance of the developed model needs to be tested further under adverse weather conditions.

Future research will focus on enriching the custom dataset through the inclusion of more nighttime and adverse weather traffic images, investigating the sensitivity of tracking hyperparameters, and integrating the framework with signal control and edge computing platforms for large-scale smart city deployment.

Data Availability: The custom dataset used in this study is currently not publicly available because it forms part of the author's ongoing research work. The dataset will be made available from the corresponding author upon reasonable request after completion of the research.

REFERENCES

[1] E. Dilek and M. Dener, "Computer Vision Applications in Intelligent Transportation Systems: A Survey," Sensors, vol. 23, no. 6, 2023, doi: 10.3390/s23062938.

[2] H. Shokravi, H. Shokravi, N. Bakhary, M. Heidarrezaei, S. S. R. Koloor, and M. Petrů, "A review on vehicle classification and potential use of smart vehicle-assisted techniques," Sensors (Switzerland), vol. 20, no. 11, pp. 1–29, 2020, doi: 10.3390/s20113274.

[3] N. A. Mandellos, I. Keramitsoglou, and C. T. Kiranoudis, "Expert Systems with Applications A background subtraction algorithm for detecting and tracking vehicles," Expert Syst. Appl., vol. 38, no. 3, pp. 1619–1631, 2011, doi: 10.1016/j.eswa.2010.07.083.

[4] X. Cao, C. Wu, P. Yan, and X. Li, "Linear SVM classification using boosting HOG features for vehicle detection in low-altitude airborne videos," Proc. - Int. Conf. Image Process. ICIP, pp. 2421–2424, 2011, doi: 10.1109/ICIP.2011.6116132.

[5] T. Azfar, J. Li, H. Yu, R. L. Cheu, Y. Lv, and R. Ke, "Deep Learning-Based Computer Vision Methods for Complex Traffic Environments Perception: A Review," Data Sci. Transp., vol. 6, no. 1, pp. 1–19, 2024, doi: 10.1007/s42421-023-00086-7.

[6] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., pp. 580–587, 2014, doi: 10.1109/CVPR.2014.81.

[7] R. Girshick, "Fast R-CNN," Proc. IEEE Int. Conf. Comput. Vis., vol. 2015 Inter, pp. 1440–1448, 2015, doi: 10.1109/ICCV.2015.169.

[8] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," IEEE Trans. Pattern Anal. Mach. Intell., vol. 39, no. 6, pp. 1137–1149, 2017, doi: 10.1109/TPAMI.2016.2577031.

[9] W. Liu et al., "SSD: Single Shot MultiBox Detector," ECCV, vol. 1, pp. 21–37, 2016, doi: 10.1007/978-3-319-46448-0_2.

[10] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., vol. 2016-Decem, pp. 779–788, 2016, doi: 10.1109/CVPR.2016.91.

[11] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017, vol. 2017-Janua, pp. 6517–6525, 2017, doi: 10.1109/CVPR.2017.690.

[12] J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement," 2018, [Online]. Available: http://arxiv.org/abs/1804.02767

[13] C. Wang and H. M. Liao, "YOLOv4: Optimal Speed and Accuracy of Object Detection".

[14] A. Farid, F. Hussain, K. Khan, M. Shahzad, U. Khan, and Z. Mahmood, "A Fast and Accurate Real-Time Vehicle Detection Method Using Deep Learning for Unconstrained Environments," Appl. Sci., vol. 13, no. 5, 2023, doi: 10.3390/app13053059.

[15] T. Vu, H. N. Thai, V. N. Pham, H. T. Vu, A. T. Luong, and T. Van Luong, "Counting Mixed Traffic Volumes at Motorcycle-Dominated Intersections by Using Computer Vision," Int. J. Intell. Transp. Syst. Res., vol. 23, no. 1, pp. 146–164, 2025, doi: 10.1007/s13177-024-00442-z.

[16] M. Safaldin, N. Zaghden, and M. Mejdoub, "An Improved YOLOv8 to Detect Moving Objects," IEEE Access, vol. 12, no. May, pp. 59782–59806, 2024, doi: 10.1109/ACCESS.2024.3393835.

[17] J. Terven, D. M. Córdova-Esparza, and J. A. Romero-González, "A Comprehensive Review of YOLO Architectures in Computer Vision: From YOLOv1 to YOLOv8 and YOLO-NAS," Mach. Learn. Knowl. Extr., vol. 5, no. 4, pp. 1680–1716, 2023, doi: 10.3390/make5040083.

[18] A. Vijayakumar and S. Vairavasundaram, YOLO-based Object Detection Models: A Review and its Applications, vol. 83, no. 35. Springer US, 2024. doi: 10.1007/s11042-024-18872-y.

[19] A. A. Murat and M. S. Kiran, "A comprehensive review on YOLO versions for object detection," Eng. Sci. Technol. an Int. J., vol. 70, no. June, 2025, doi: 10.1016/j.jestch.2025.102161.

[20] X. Duan, S. Xue, Z. Zhou, and X. Duan, "Design and Implementation of Road Vehicle Counting Model Based on YOLOv8," Proc. 2024 8th Int. Conf. Electron. Inf. Technol. Comput. Eng. EITCE 2024, pp. 844–848, 2025, doi: 10.1145/3711129.3711275.

[21] W. Ai, J. Zeng, Z. Lei, and D. Li, "Traffic flow parameter estimation for complex environments based on improved YOLOv8," Proc. Inst. Mech. Eng. Part D J. Automob. Eng., vol. 239, no. 12, pp. 5412–5426, 2025, doi: 10.1177/09544070241287582.

[22] M. Bakirci, "Utilizing YOLOv8 for enhanced traffic monitoring in intelligent transportation systems (ITS) applications," Digit. Signal Process. A Rev. J., vol. 152, no. May, p. 104594, 2024, doi: 10.1016/j.dsp.2024.104594.

[23] B. Ji, J. Xu, Y. Liu, P. Fan, and M. Wang, "Improved YOLOv8 for small traffic sign detection under complex environmental conditions," Franklin Open, vol. 8, no. October, p. 100167, 2024, doi: 10.1016/j.fraope.2024.100167.

[24] Iqra and K. J. Giri, "SO-YOLOv8: A novel deep learning-based approach for small object detection with YOLO beyond COCO," Expert Syst. Appl., vol. 280, no. March, p. 127447, 2025, doi: 10.1016/j.eswa.2025.127447.

[25] H. Li, Y. Li, L. Xiao, Y. Zhang, L. Cao, and D. Wu, "RLRD-YOLO: An Improved YOLOv8 Algorithm for Small Object Detection from an Unmanned Aerial Vehicle (UAV) Perspective," Drones, vol. 9, no. 4, pp. 1–21, 2025, doi: 10.3390/drones9040293.

[26] M. Bakirci, "Advanced aerial monitoring and vehicle classification for intelligent transportation systems with YOLOv8 variants," J. Netw. Comput. Appl., vol. 237, no. November 2023, p. 104134, 2025, doi: 10.1016/j.jnca.2025.104134.

[27] N. Wojke, A. Bewley, and D. Paulus, "Simple online and realtime tracking with a deep association metric," Proc. - Int. Conf. Image Process. ICIP, vol. 2017-Septe, pp. 3645–3649, 2017, doi: 10.1109/ICIP.2017.8296962.

[28] H. Song, H. Liang, H. Li, Z. Dai, and X. Yun, "Vision-based vehicle detection and counting system using deep learning in highway scenes," Eur. Transp. Res. Rev., vol. 11, no. 1, 2019, doi: 10.1186/s12544-019-0390-4.

[29] M. Durve, S. Orsini, A. Tiribocchi, A. Montessori, and J. Tucny, "Benchmarking YOLOv5 and YOLOv7 models with DeepSORT for droplet tracking applications," Eur. Phys. J. E, vol. 46, no. 5, pp. 1–7, 2023, doi: 10.1140/epje/s10189-023-00290-x.

[30] B. Sathiyaprasad, P. A. R. Reddy, and D. M. Venkatesh, "TrackNCount - Intelligent Vehicle Tracking, Counting, and Speed Estimation using YOLOv8 and DeepSORT Algorithms," 6th Int. Conf. Mob. Comput. Sustain. Informatics, ICMCSI 2025 - Proc., pp. 1738–1744, 2025, doi: 10.1109/ICMCSI64620.2025.10883110.

[31] M. R. Sholahuddin et al., "Optimizing YOLOv8 for Real-Time CCTV Surveillance: A Trade-off Between Speed and Accuracy," J. Online Inform., vol. 8, no. 2, pp. 261–270, 2023, doi: 10.15575/join.v8i2.1196.

[32] A. Akoushideh, S. S. Sadat, and A. Shahbahrami, "Counting vehicles types using deep learning algorithm in video surveillance systems," Multimed. Tools Appl., vol. 84, no. 27, pp. 33017–33040, 2025, doi: 10.1007/s11042-024-20530-2.

[33] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (VOC) challenge," Int. J. Comput. Vis., vol. 88, no. 2, pp. 303–338, 2010, doi: 10.1007/s11263-009-0275-4.

[34] G. T. U. A. Colleges et al., "Microsoft COCO," Eccv, no. June, pp. 740–755, 2014.

[35] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The KITTI dataset," Int. J. Rob. Res., vol. 32, no. 11, pp. 1231–1237, 2013, doi: 10.1177/0278364913491297.

[36] M. Cordts et al., "The Cityscapes Dataset for Semantic Urban Scene Understanding," Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., vol. 2016-Decem, pp. 3213–3223, 2016, doi: 10.1109/CVPR.2016.350.

[37] A. Kuznetsova et al., "The Open Images Dataset V4: Unified Image Classification, Object Detection, and Visual Relationship Detection at Scale," Int. J. Comput. Vis., vol. 128, no. 7, pp. 1956–1981, 2020, doi: 10.1007/s11263-020-01316-z.

[38] B. A. Paranjape and A. A. Naik, "DATS_2022: A versatile indian dataset for object detection in unstructured traffic conditions," Data Br., vol. 43, pp. 0–6, 2022, doi: 10.1016/j.dib.2022.108470.

[39] C. Y. Wang, H. Y. Mark Liao, Y. H. Wu, P. Y. Chen, J. W. Hsieh, and I. H. Yeh, "CSPNet: A new backbone that can enhance learning capability of CNN," IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Work., vol. 2020-June, pp. 1571–1580, 2020, doi: 10.1109/CVPRW50498.2020.00203.

[40] Z. Zheng, P. Wang, W. Liu, J. Li, R. Ye, and D. Ren, "Distance-IoU loss: Faster and better learning for bounding box regression," AAAI 2020 - 34th AAAI Conf. Artif. Intell., no. 2, pp. 12993–13000, 2020, doi: 10.1609/aaai.v34i07.6999.

[41] X. Li et al., "Generalized focal loss: Learning qualified and distributed bounding boxes for dense object detection," Adv. Neural Inf. Process. Syst., vol. 2020-Decem, pp. 1–14, 2020.