IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received 7 November 2024, accepted 2 January 2025, date of publication 7 January 2025, date of current version 15 January 2025.

Digital Object Identifier 10.1109/ACCESS.2025.3526880

RESEARCH ARTICLE

Vehicle Turn Pattern Counting and Short Term Forecasting Using Deep Learning for Urban Traffic Management System

SUNDARAKRISHNAN NARAYANAN[1], (Student Member, IEEE), SOHAN VARIER[1], TARUN BHUPATHI[1], (Student Member, IEEE), MANASWINI SIMHADRI KAVALI[1], MOHANA[1], P. RAMAKANTH KUMAR[1], (Senior Member, IEEE), AND K. SREELAKSHMI[2], (Senior Member, IEEE)

[1]Department of Computer Science and Engineering, R. V. College of Engineering, Bengaluru 560059, India
[2]Department of Electronics and Telecommunication Engineering, R. V. College of Engineering, Bengaluru 560059, India

Corresponding author: Sundarakrishnan Narayanan (sundarakn.cs22@rvce.edu.in)

This work was supported by The Center for Computer Vision Research, R. V. College of Engineering.

ABSTRACT Urban traffic management has been facing increasing challenges due to the surge in number of vehicles and traffic congestion. As cities expand and population grows, efficient and accurate monitoring of vehicle counts is crucial for better traffic management. Hence, as part of the Bengaluru Mobility Challenge 2024, organized by Bengaluru Traffic Police in collaboration with The Indian Institute of Science, we propose a solution to address the issue by developing a predictive model to estimate vehicle counts by turning pattern from traffic video footage. The dataset consists of traffic video footage of 23 different junctions around Bengaluru, on which 7 vehicle classes had to be detected, namely, Car, Truck, Bus, Two-Wheeler, Three-Wheeler, Light Commercial Vehicle and Bicycle. The proposed work focuses on two key objectives: counting vehicle turns over 30-minute clips and forecasting future vehicle turn counts by class for the next 30 minutes. A You Only Look Once (YOLOv8) and Auto-ARIMA based pipeline was deployed to address the challenge, which demonstrated robust detection capabilities, with an overall precision of 92.59% for vehicle detection. Building on this, we designed a custom vehicle counting algorithm that integrated the BoT-SORT tracker with dynamic counting boxes, accurately capturing vehicle movements and turn patterns in real-time and this integrated approach attained a best deviation of 20.79%. for turn pattern counting and 28.41% for forecasting. Furthermore, the system is scalable to accommodate any number of cameras and is capable of forecasting traffic over extended time frames, allowing it to be applied to a variety of urban traffic monitoring scenarios. These results highlight the effectiveness of our custom designed framework in real-world scenarios as a reliable model for applications needing high-precision detection and predictive analytics.

INDEX TERMS Auto-ARIMA, deep learning, object tracking, time-series analysis, traffic forecasting, YOLOv8, urban traffic management, vehicle turn pattern counting.

I. INTRODUCTION

Traffic on Indian roads has been increasing exponentially over the past years, primarily due to rapid urbanization and the rise in vehicle ownership. This unprecedented surge in vehicular traffic has placed immense pressure on existing road infrastructure in metropolitan areas like Bengaluru, India, and has mandated a change from the traditional, manual method of traffic management to smart, automated systems that efficiently navigate vehicular traffic at critical junctions across a city. This has led to vast research and innovation in the field of developing smart traffic management systems. The development of smart traffic management systems (STMS) has emerged as a pivotal solution to address

The associate editor coordinating the review of this manuscript and approving it for publication was Sajid Ali.

these challenges. These systems use advanced technologies such as Artificial Intelligence (AI), Machine Learning (ML), Computer Vision, and the Internet of Things (IoT) to analyze real-time traffic data and optimize traffic flow. One of the key aspects of these systems is their ability to predict and forecast traffic patterns, which can enable proactive interventions such as dynamic signal control, congestion mitigation, and route optimization.

This paper focuses on a critical component of smart traffic management, predicting vehicle counts by class in real-time from camera feeds. The ability to predict vehicle counts is essential not only for traffic management, but also for urban planning [20]. Specifically, this research addresses two key problems. The first is counting the number of turns made by vehicles for 7 classes (cars, buses, trucks, two-wheelers, three-wheelers, bicycles and LCV) observed from the start to end of a 30-minute video clip. Additionally, the solution implemented in the paper forecasts the vehicle turn pattern count for the subsequent 30 minutes, enabling traffic systems to anticipate and adjust to future traffic flows. Another challenge lies in generalizing this model to unseen locations. Traffic conditions in different parts of a city can vary significantly due to factors like road design, local population density, and the presence of commercial or residential areas. Hence, it is important to develop a model that not only performs well in known locations, but also has the ability to predict vehicle counts at previously unseen locations. This aspect of the study enhances the model's adaptability and scalability across the city's diverse traffic environments.

By addressing these challenges, this work aims to demonstrate the model's ability to generalize across different traffic scenarios and provide accurate predictions in both familiar and unfamiliar environments. This research has significant implications for improving traffic flow in major urban cities like Bengaluru, where congestion is a daily problem. The insights gained from this work could be used to design smarter traffic systems that dynamically respond to traffic patterns, leading to reduced delays and eventually contributing to a more sustainable urban ecosystem.

In summary, this research explores the intersection of machine learning, forecasting, and traffic management, offering a novel solution to the growing issue of traffic congestion in cities like Bengaluru. While numerous smart traffic management systems exist globally, there is a notable lack of robust solutions tailored to India-specific scenarios, and this study addresses that gap. The predictive model developed as a part of the Bengaluru Mobility Challenge [1], demonstrated in this paper provides a foundation for smarter, more responsive traffic systems that not only manage current traffic but also anticipate future challenges, enabling urban environments to function more efficiently in the face of rising vehicular pressure. The framework leverages technologies like You Only Look Once (YOLOv8) [12], and Auto-ARIMA (Auto-Regressive Integrated Moving Average) [13] to effeciently solve the problem.

The rest of the paper is organized as follows: Section II consists of the Literature Survey, Section III consists of the Design and Implementation, and Section IV describes the Results. Section V contains the conclusion.

II. LITERATURE SURVEY

Folenta et al. [2] created a vehicle turn counting framework using YOLOv3 for vehicle detection and DeepSORT for vehicle tracking. A trajectory based system was used to determine the vehicle turn pattern. However, in Indian road conditions, the trajectory may not be very well defined or fixed. Gloudemans et al. [3] worked on a fast vehicle turn counting design to work on edge computing devices. Detection was performed on partial cropped frames to increase performance. However, this may not be very effective in dense traffic conditions. Shirazi et al. [4] introduced a vision-based vehicle tracking system to count turning movements at intersections using two complementary modules. One ensured robust tracking, and the other used trajectory comparisons to address occlusions. This system may struggle in high dynamic environments with frequent changes, impacting its efficiency in real-time applications.

He et al. [5] proposed the use of roadside radars, along with camera imagery, to predict vehicle turn intentions before turning. The system made use of an LSTM-GAN framework. Abbas et al. [6] proposed a comparison between three different models for short-term traffic density, all of which were based on Long Short-Term Memory (LSTM) neural networks. The models were trained on real traffic data collected in Stockholm. However, this relied on the deployment of a full set of sensors, which may be impractical in real-world scenarios. Alzyout et al. [7] developed an automated GPS location prediction system using a modified ARIMA model. Evaluations using real-time data showed accurate predictions within a reasonable time frame. For tighter deadlines, the last ready model was used to predict the GPS location of the vehicle. Tran et al. [8] presents TsModeler, an automatic time series modeling framework that uses ARIMA and spatial Markov models to predict temporal I/O patterns for adaptive prefetching in high-performance computing systems. While the performance improved, one major shortcoming was the potential overhead introduced by the sheer amount of computing resources required for real-time model identification, which may in turn limit scalability in large-scale and highly dynamic systems such as the case of predicting real-time traffic in the roads of a city like Bengaluru. Shuvo et al. [9] conducted a study to compare various Time-Series analysis models (ARIMA, ETS, SNAIVE, PROPHET and a mix of all), based on traffic in Dhaka, Bangladesh. Roads showing seasonality were best forecast by PROPHET, whereas in SNAIVE worked best for roads with less seasonality. Overall PROPHET performed well, achieving an accuracy of 91%. Hossain et al. [10] in the paper, investigated traffic prediction methods for Dhaka City using machine learning and deep learning techniques, comparing models such as Bagged trees, SVM, and LSTM. The study demonstrated that LSTM provided the highest accuracy for traffic forecasting. A major shortcoming was the dataset's limitations, which could delay the deep learning model's implementation and may hinder the achievement of the full potential of more advanced methods like neural networks. Hardjono et al. [11] evaluated vehicle counts on four datasets from Indonesian highways. A Deep Learning method with YOLO provided the best results thus proving its efficiency in traffic systems.

III. DESIGN AND IMPLEMENTATION

An efficient methodology pipeline was devised as per the needs of the challenge as shown in Fig.1. First, vehicle counts by class along with their turning pattern had to be counted for 30 minutes, along with short-term forecasting of traffic for the next 30 minutes across all classes and turn patterns. The videos from the dataset were split into frames to perform model training. Next, a turn pattern counting algorithm was devised and tested across multiple cameras. Data that was logged during this 30 minutes was preprocessed for forecasting using Auto-ARIMA.

A. DATA PREPROCESSING

As a part of the challenge, video feed across 23 safe city cameras in Bengaluru was provided at varying times of a day for an entire week. From this dataset, specific videos were selected and frames were extracted from each selected video at a rate of one frame every two seconds. Camera junctions were strategically chosen at different times to ensure a diverse representation of scenarios and to enhance the variety of data collected.

The first task was to develop a basic YOLOv8 model to detect seven vehicle classes: Car, Bus, Two-wheeler, Three-wheeler, Light Commercial Vehicles (LCV), Truck and Bicycle. Given the labor-intensive and time consuming nature of manual annotation, an auto-annotation framework was employed to expedite the process. Initially, a baseline model was trained on a manually annotated subset of the dataset. This model was then used to automatically annotate additional images, followed by manual review and correction to ensure annotation quality.

The iterative process involved 12 cycles of annotation and training, with each cycle resulting in improved model performance and annotation accuracy. During each iteration, the dataset consisted of strategically chosen junctions which are potentially detrimental to the model's accuracy, allowing for focus on areas that required improvement. To address underrepresented classes, such as Bicycles, data augmentation techniques were applied to provide requisite images for training. Upon completion of the iterative process, the final dataset comprised 8,000 images for training and 800 images each for testing and validation. Fig.2 shows the the dataset's vehicle type distribution with Two-wheelers being the most common, followed by Cars and Three-wheelers.

The entire pipeline was designed to seamlessly integrate auto-annotation, manual review and model retraining. At each iteration, when new data was added, key metrics such as confusion matrices were analyzed to assess model performance. A detailed discussion of vehicle turn pattern counting and forecasting is provided in subsequent sections.

B. CREATING COUNTING BOXES

A utility script was created to help configure the turn patterns for different camera views. A snapshot of the relevant camera feed is opened, where the turning regions are defined by clicking on four points to create a polygon in the desired region. The coordinates have been chosen to maximize the coverage of area, reducing the possibility of a vehicle not being detected. All the coordinates are then stored in a dictionary that links camera names and their corresponding regions of interest as key value pairs. This facilitated easier referencing and modification of the turning regions, thus ensuring greater adaptability and scalability.

C. VEHICLE TURN PATTERN COUNTING

The challenge provided the valid turn pattern configurations for each camera and we were tasked to count and forecast the turns made by each class of vehicle. The important task after vehicle detection was accurate vehicle tracking. The BoT-SORT [14] tracking algorithm was used with the YOLOv8 tracker to assign each vehicle a unique ID. This approach ensured consistency of vehicle IDs across frames even in dense traffic conditions due to its ability to combine the benefits of motion and appearance information, camera-motion compensation, and an enhanced Kalman filter. YOLO's polygon object counter was utilized to make detections, and the logic of counting only the valid turns was developed.. YOLOv8's object counter works by counting all the vehicles detected within a manually annotated polygon region. The counter was further modified based on the need to log more data needed for counting turn patterns, where different turning regions were assigned different boxes to estimate and detect vehicle movement from one region to another. Taking the STN HD 1 junction as an example, as given in Fig.4, there are 6 possible turns; BC, BE, DA, DE, FA and FE.

When a vehicle is initially detected by the YOLOv8 model, a unique ID is assigned to it. Upon its first detection within a specific region/box, the system checks for the ID in a global dictionary. If the ID is not found, the unique ID and its corresponding counter box are designated as the source, indicating it was not previously detected in any other box. When the same vehicle is detected in another box, the ID is cross-checked against the dictionary. If the ID is found, the corresponding box is added as a destination box for that ID. The config file also houses the possible turn patterns for that particular junction, and using this it matches the source and

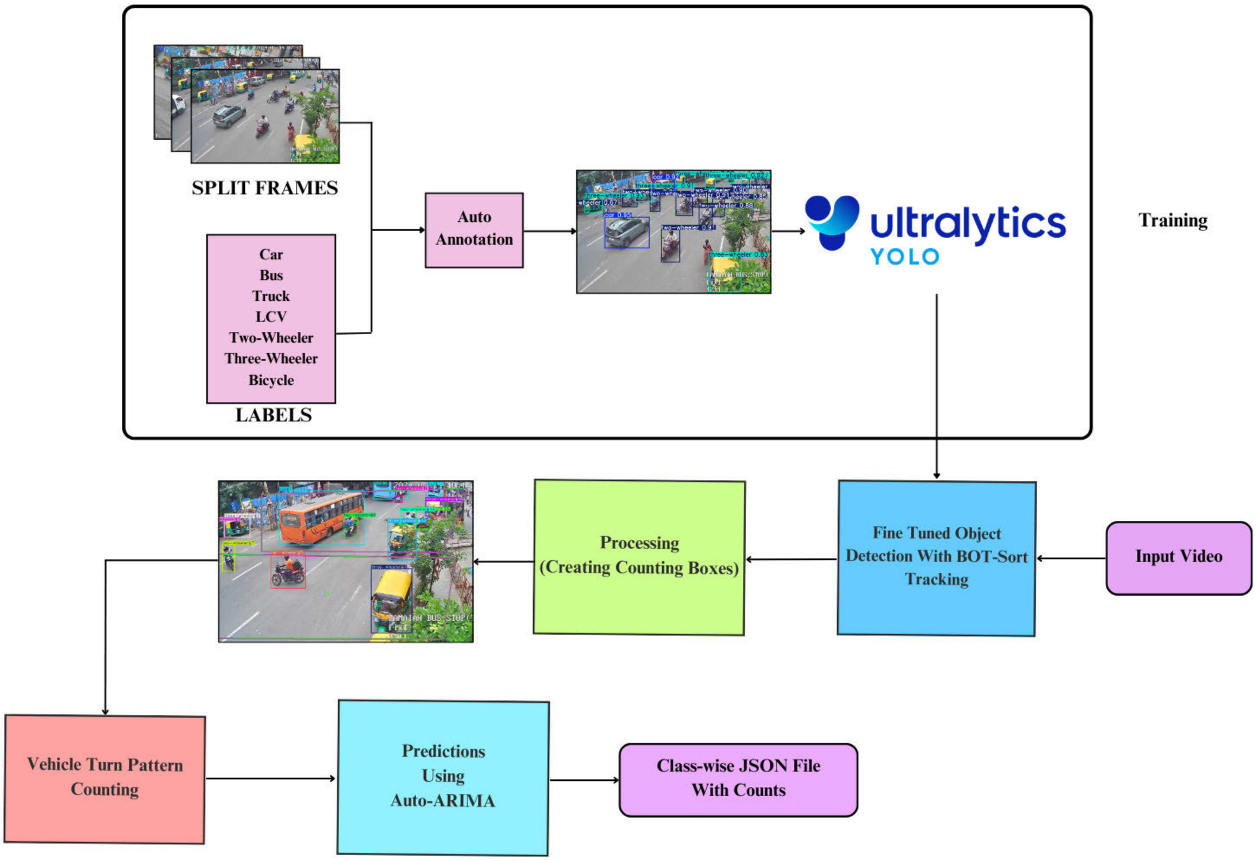


FIGURE 1. Vehicle detection and counting pipeline using YOLOv8, BoT-SORT tracking, and Auto-ARIMA for traffic prediction.

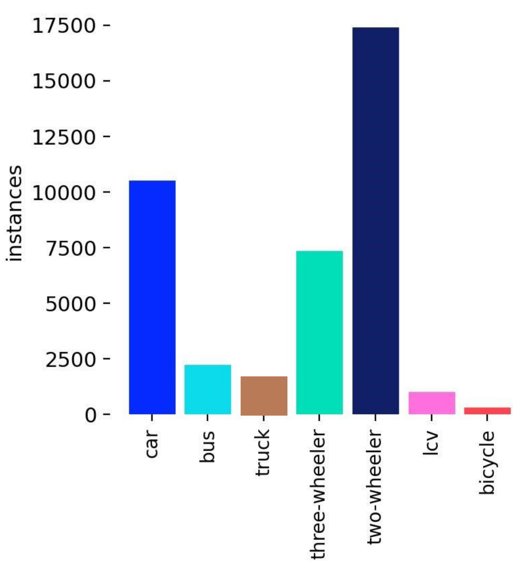


FIGURE 2. Dataset distribution by vehicle class.

FIGURE 3. Sample annotated image from dataset.

destination boxes to the resulting turn pattern. Each camera had a different set of regions/boxes with different valid turn patterns depending on the junction and the information of the coordinates and turn patterns were saved in a config file. The naive approach would be to create six regions and check for turns between them. Upon testing, it was observed that many vehicles were making turns while cutting through other regions, which gave wrong counts, eg. when turning from F to C, many vehicles were found going through E and D too because of poor lane discipline. A solution to this was to merge the boxes of each side, and then map the source and destination boxes to turn patterns based on the junction. In this scenario, A and B, C and D and E and F were merged

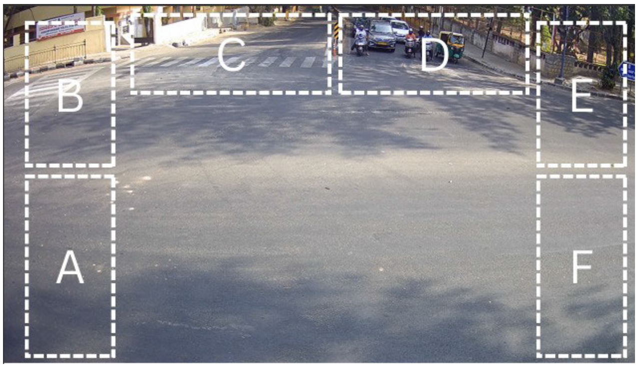


FIGURE 4. The valid turn patterns would be BC, BE, DA, DE, FA and FE.

FIGURE 5. Updated combined junctions to tackle the problem of poor lane discipline.

to form three boxes J1, J2 and J3 respectively, as shown in Fig.5. When a turn was detected with J1 as source and J2 as destination, it would be counted as a BC turn, as it is known that AD is not a valid turning pattern. These mappings are also present in the config file. An example config file looked as follows:

```
'Stn_HD_1':
{
'regions': {
'J1': [(0, 0), (180, 0), (180, 810), (0,
    810)],
'J2': [(180, 0), (1260, 0), (1260, 130),
    (180, 130)],
'J3': [(1260, 0), (1440, 0), (1440, 810)
    , (1260, 810)],
},
'turning_patterns': {
'BC': ['J1', 'J2'],
'BE': ['J1', 'J3'],
'DA': ['J2', 'J1'],
'DE': ['J2', 'J3'],
'FA': ['J3', 'J1'],
'FC': ['J3', 'J2'],
}
}
```

Another addition was in cases where a rider moved from J1 to J3 through J2 like in Fig.5 or any such similar patterns in other junctions,then only the pattern J3 is considered. In summary, if a vehicle passed through multiple destination boxes, only the final valid destination would be counted, since the intermediate ones were logged either due to poor camera position or lane discipline. The entire system is scalable and can be calibrated for new cameras simply by defining the boxes and the valid turn patterns.

D. FORECASTING

A framework for forecasting vehicle counts at various traffic junctions using the Auto-ARIMA(AutoRegressive Integrated Moving Average) model was developed. Given the dynamic nature of Indian roads with high variability from location to location and other factors like time, a time series model with fixed hyper parameters could not be used. Auto-ARIMA was used to dynamically find the optimal parameters for forecasting depending on the data logged, thereby increasing accuracy considerably.

The framework consists of several key steps, starting with data collection and preprocessing. Initially, vehicle counts were logged over time along with the frame numbers where the vehicle executed a specific turning pattern. This provided time-stamped data required for forecasting. The time-stamped data was then preprocessed, where it was organized and interpolated to align with specific time intervals, ensuring consistency and effective analysis of trends across time intervals. Auto-ARIMA automates the selection of the optimal combination of ARIMA model parameters (p, d, q) by minimizing information criteria such as AIC or BIC, ensuring that the model is efficient. The model was applied to each vehicle class across different traffic turns, utilizing vehicle counts from the previous 30 minutes to forecast the same for the next 30 minutes. Predicted values were then compared to the actual observations to evaluate the model's performance. Fig.6 and Fig.7 are some examples of forecasted counts for cars and trucks making the BC turn at STN HD 1 camera. An ARIMA model can be defined using the following equation:

$$y_t = c + \sum_{i=1}^{p} \phi_i y_{t-i} + \sum_{j=1}^{q} \theta_j \varepsilon_{t-j} + \varepsilon_t \tag{1}$$

where:

- y_t is the value of the time series at time t.
- c is a constant term.
- ϕ_i are the coefficients of the autoregressive (AR) terms, where $i = 1, 2, \ldots, p$.
- θ_j are the coefficients of the moving average (MA) terms, where $j = 1, 2, \ldots, q$.
- ε_t is the error term at time t, assumed to be a white noise process.
- p is the number of lag observations of the model.
- q(order of moving average) is the size of the moving average window.

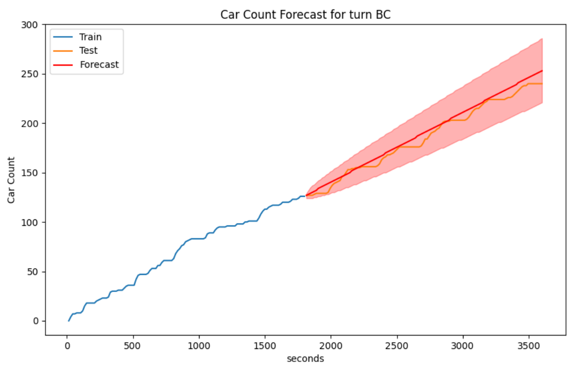


FIGURE 6. Predicted cumulative turns for cars making the BC turn.

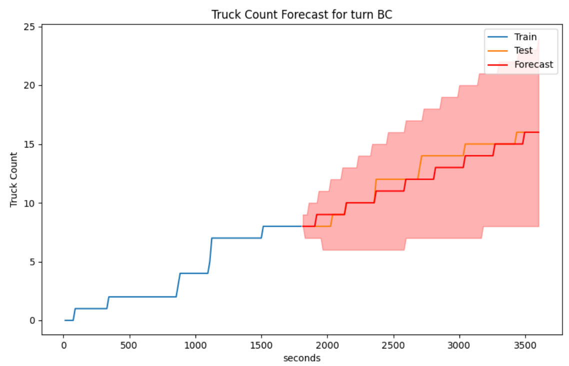


FIGURE 7. Predicted cumulative turns for trucks making the BC turn.

- d is the degree of differencing needed to make the series stationary. If $d > 0$, then differencing is applied d times to the original series.

The main parameters are p, q and d that need to be optimized for the best results.

AIC (Akaike Information Criterion) and BIC (Bayesian Information Criterion) are statistical results that are used to evaluate the quality of a model and also penalize its complexity. This allowed the model to strike a balance between accuracy and simplicity. Auto-ARIMA makes use of these measures to optimize the ARIMA parameters to achieve the best possible results in terms of time-series predictions.

The framework offers flexibility to add additional vehicle classes without much hassle, making it a reliable and adaptive solution.

IV. RESULTS AND ANALYSIS

The performance of the designed YOLOv8 model showed significant improvement in object detection after being trained for 350 epochs. Techniques like dropout regularization and label smoothing were utilized, which helped mitigate over-fitting and improve generalization across various classes. Key evaluation metrics, such as precision and recall, were essential in quantifying the effectiveness of the model.

The specific metrics used to evaluate the model's performance are provided below:

$$\text{Precision} = \frac{\text{True Positives (TP)}}{\text{True Positives (TP)} + \text{False Positives (FP)}} \tag{2}$$

$$\text{Recall} = \frac{\text{True Positives (TP)}}{\text{True Positives (TP)} + \text{False Negatives (FN)}} \tag{3}$$

$$\text{mAP@50} = \frac{1}{N}\sum_{i=1}^{N} \text{AP}_i^{0.5} \tag{4}$$

where:

- mAP@50: Mean Average Precision at 50% Intersection over Union (IoU) threshold.
- N: Total number of object categories (classes) in the dataset.
- $\text{AP}_i^{0.5}$: Average Precision (AP) for the i-th class at an IoU threshold of 0.5. It is calculated as the area under the Precision-Recall (PR) curve for that specific class.

The model demonstrated a strong overall precision of 92.58% and a recall of 85.93%, with classes such as Cars and Buses achieving a precision of 96%. Fig.8 illustrates a confusion matrix that presents the normalized classification accuracy of the model across the classes. The Diagonal values indicate a high accuracy for Cars (96%), Bus (96%), Truck (95%), and Two-wheeler/Three-wheelers (95%). The model shows room for improvement for LCV (84%) and Bicycles (58%) due to their visual similarities with other classes and smaller size respectively. Fig.9 further illustrates the model's performance through a precision-recall curve. The high plateau in the curve demonstrates the model's ability to achieve strong precision consistently. An overall mean Average Precision (mAP) of 91.9% at 0.5 Intersection over Union (IoU) was achieved. LCV (84.6%) and Bicycles (73.3%) exhibited significant

$$d(t,v) = \begin{cases} \min\left(\dfrac{|GT_{t,v} - SC_{t,v}|}{GT_{t,v}}, 1\right) & \text{if } GT_{t,v} > 0 \\ 1 & \text{if } GT_{t,v} = 0, SC_{t,v} > 0 \\ 0 & \text{if } GT_{t,v} = 0, SC_{t,v} = 0 \end{cases} \tag{5}$$

$$\text{deviation}(t,v) = \frac{1}{|T||V|}\sum_{t\in T}\sum_{v\in V} d(t,v) \times 100\% \tag{6}$$

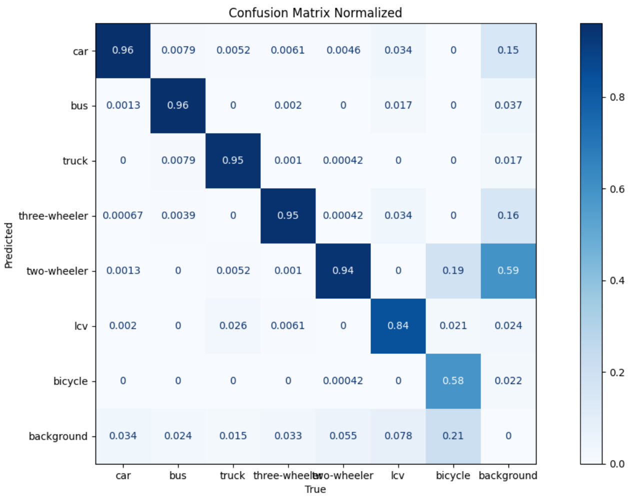


FIGURE 8. Normalized confusion matrix for YOLOv8 Model.

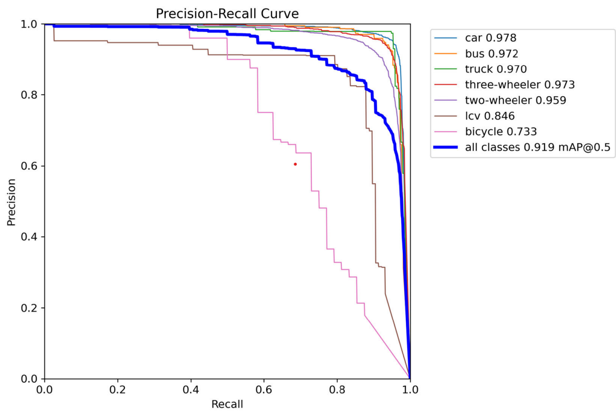


FIGURE 9. Precision-Recall curve for the model.

drops in their curves at higher recall values, suggesting that the model faces challenges in identifying smaller or visually ambiguous instances of vehicles. Overall, the high mAP and the smooth curve indicate the effectiveness of the model across most categories, making it a promising tool for practical deployment in traffic management systems.

Deviation for counting and forecasting algorithm was computed as in (5) and (6), shown at the bottom of the previous page, where:

- T is the set of all turn patterns;
- V is the set of all vehicle types;
- $GT_{t,v}$ is the ground truth vehicle count for turning pattern t and vehicle type v;
- $SC_{t,v}$ is the vehicle count output by the submitted code for turning pattern t and vehicle type v during the given 30-minute clip, as well as the predicted vehicle count for the next 30 minutes by the algorithm for turning pattern t and vehicle type v.

The overall pipeline achieved a best deviation of 20.79% for turn pattern counting and 28.41% for prediction in the competition and secured first place in The Bengaluru Mobility Challenge, 2024. The design and implementation is available on GitHub.

V. CONCLUSION

A robust counting framework has been successfully developed and has proven its accuracy across diverse road conditions in India. This framework has applications in smart traffic management, where signals adapt dynamically to projected vehicle turns, reducing congestion and enabling early detection of traffic build-ups. Further, it can support efficient traffic police deployment for traffic management and help ensure clear routes for emergency vehicles such as ambulances, fire engines, etc. and other vehicles when necessary. The proposed solution is lightweight and scalable, allowing new cameras to be added with ease. The pipeline achieved a best count deviation of 20.79% and a forecast deviation of 28.41%, securing First Place in The IEEE Bengaluru Mobility Challenge 2024. Further research can be done to address the issue of occlusions, where a vehicle may not be detected due to a larger vehicle overlapping it, based on the camera angle. Abnormal traffic conditions such as road blockages due to VIP movements or accidents may also cause a deviation from the predicted count values and produce inaccuracies.

ACKNOWLEDGMENT

The authors would like to thank the organizers of Bengaluru Mobility Challenge for providing the dataset. The above work secured the First Place in the Bengaluru Mobility Challenge, in 2024.

REFERENCES

[1] "The Bengaluru mobility challenge, 2024," IEEE Dataport, Bengaluru Traffic Police, Tech. Rep., Jun. 2024, doi: 10.21227/8ey0-sw97.

[2] J. Folenta, J. Špaňhel, V. Bartl, and A. Herout, "Determining vehicle turn counts at multiple intersections by separated vehicle classes using CNNs," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Seattle, WA, USA, Jun. 2020, pp. 2544–2549, doi: 10.1109/CVPRW50498.2020.00306.

[3] D. Gloudemans and D. B. Work, "Fast vehicle turning-movement counting using localization-based tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Nashville, TN, USA, Jun. 2021, pp. 4150–4159, doi: 10.1109/CVPRW53098.2021.00469.

[4] M. S. Shirazi and B. Morris, "Vision-based turning movement counting at intersections by cooperating zone and trajectory comparison modules," in *Proc. 17th Int. IEEE Conf. Intell. Transp. Syst. (ITSC)*, Qingdao, China, Oct. 2014, pp. 3100–3105, doi: 10.1109/ITSC.2014.6958188.

[5] J.-H. He, Y.-L. Chen, X.-Z. Chen, and H.-H. Chiang, "Vehicle turning intention prediction based on data-driven method with roadside radar and vision sensor," in *Proc. IEEE Int. Conf. Consum. Electronics-Taiwan (ICCE-TW)*, Taiwan, Sep. 2021, pp. 1–2, doi: 10.1109/ICCE-TW52618.2021.9602976.

[6] Z. Abbas, A. Al-Shishtawy, S. Girdzijauskas, and V. Vlassov, "Short-term traffic prediction using long short-term memory neural networks," in *Proc. IEEE Int. Congr. Big Data (BigData Congress)*, San Francisco, CA, USA, Jul. 2018, pp. 57–65, doi: 10.1109/BIGDATACONGRESS.2018.00015.

[7] M. Alzyout, M. Alsmirat, and M. I. Al-Saleh, "Automated ARIMA model construction for dynamic vehicle GPS location prediction," in *Proc. 6th Int. Conf. Internet Things, Syst., Manage. Secur. (IOTSMS)*, Granada, Spain, Oct. 2019, pp. 380–386, doi: 10.1109/IOTSMS48152.2019.8939197.

[8] N. Tran and D. A. Reed, "Automatic ARIMA time series modeling for adaptive I/O prefetching," *IEEE Trans. Parallel Distrib. Syst.*, vol. 15, no. 4, pp. 362–377, Apr. 2004, doi: 10.1109/tpds.2004.1271185.

[9] M. A. R. Shuvo, M. Zubair, A. T. Purnota, S. Hossain, and M. I. Hossain, "Traffic forecasting using time-series analysis," in *Proc. 6th Int. Conf. Inventive Comput. Technol. (ICICT)*, Coimbatore, India, Jan. 2021, pp. 269–274, doi: 10.1109/ICICT50816.2021.9358682.

[10] M. N. Hossain, N. Ahmed, and S. M. W. Ullah, "Traffic flow forecasting in intelligent transportation systems prediction using machine learning," in *Proc. Int. Conf. Futuristic Technol. (INCOFT)*, Belagavi, India, Nov. 2022, pp. 1–5, doi: 10.1109/INCOFT55651.2022.10094319.

[11] B. Hardjono, H. Tjahyadi, M. G. A. Rhizma, A. E. Widjaja, R. Kondorura, and A. M. Halim, "Vehicle counting quantitative comparison using background subtraction, Viola Jones and deep learning methods," in *Proc. IEEE 9th Annu. Inf. Technol., Electron. Mobile Commun. Conf. (IEMCON)*, Vancouver, BC, Canada, Nov. 2018, pp. 556–562, doi: 10.1109/IEMCON.2018.8615085.

[12] G. Jocher, J. Qiu, and A. Chaurasia. (2023). *Ultralytics YOLO (Version 8.0.0)*. [Online]. Available: https://github.com/ultralytics/ultralytics

[13] T. G. Smith. (2023). *PMDARIMA (Version 2.0.4) [Computer Software]*. [Online]. Available: https://github.com/alkaline-ml/pmdarima

[14] N. Aharon, R. Orfaig, and B.-Z. Bobrovsky, "BoT-SORT: Robust associations multi-pedestrian tracking," 2022, *arXiv:2206.14651*.

[15] B. T. Morris and M. M. Trivedi, "Learning, modeling, and classification of vehicle track patterns from live video," *IEEE Trans. Intell. Transp. Syst.*, vol. 9, no. 3, pp. 425–437, Sep. 2008, doi: 10.1109/tits.2008.922970.

[16] D. Beymer, P. McLauchlan, B. Coifman, and J. Malik, "A real-time computer vision system for measuring traffic parameters," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, San Juan, PR, USA, Jun. 1997, pp. 495–501, doi: 10.1109/CVPR.1997.609371.

[17] A. Kumar, R. Chandel, and M. Singh, "Image processing technique for tracking and counting of vehicles using ROI," in *Proc. Int. Conf. Power Energy, Environ. Intell. Control (PEEIC)*, Greater Noida, India, Dec. 2023, pp. 981–986, doi: 10.1109/peeic59336.2023.10451264.

[18] C. S. Asha and A. V. Narasimhadhan, "Vehicle counting for traffic management system using YOLO and correlation filter," in *Proc. IEEE Int. Conf. Electron., Comput. Commun. Technol. (CONECCT)*, Bengaluru, India, Mar. 2018, pp. 1–6, doi: 10.1109/CONECCT.2018.8482380.

[19] H. Hasri, S. A. Mohd Aris, and R. Ahmad, "Comparison of auto ARIMA and auto SARIMA performance in COVID-19 prediction," in *Proc. IEEE 2nd Nat. Biomed. Eng. Conf. (NBEC)*, Melaka, Malaysia, Sep. 2023, pp. 106–110, doi: 10.1109/NBEC58134.2023.10352616.

[20] A. D. Bharade and S. S. Gaopande, "Robust and adaptive traffic surveillance system for urban intersections on embedded platform," in *Proc. Annu. IEEE India Conf. (INDICON)*, Pune, India, Dec. 2014, pp. 1–5, doi: 10.1109/INDICON.2014.7030429.

[21] Q. Yu, L. Jibin, and L. Jiang, "An improved ARIMA-based traffic anomaly detection algorithm for wireless sensor networks," *Int. J. Distrib. Sensor Netw.*, vol. 12, no. 1, Jan. 2016, Art. no. 9653230, doi: 10.1155/2016/9653230.

[22] B. G. Rajagopal, N. Vishakraj, N. U. Kumar, and P. Jothivenkatesh, "Vision-based system for counting of moving vehicles in different weather conditions," in *Proc. Int. Conf. Electron., Commun. Aerosp. Technol. (ICECA)*, vol. 1, Coimbatore, India, Apr. 2017, pp. 86–91, doi: 10.1109/ICECA.2017.8203649.

[23] N. K. Kanhere and S. T. Birchfield, "Real-time incremental segmentation and tracking of vehicles at low camera angles using stable features," *IEEE Trans. Intell. Transp. Syst.*, vol. 9, no. 1, pp. 148–160, Mar. 2008, doi: 10.1109/TITS.2007.911357.

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

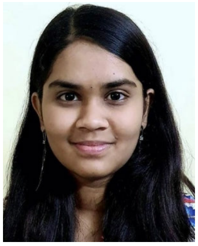
[25] Y. Gurwicz, R. Yehezkel, and B. Lachover, "Multiclass object classification for real-time video surveillance systems," *Pattern Recognit. Lett.*, vol. 32, no. 6, pp. 805–815, Apr. 2011, doi: 10.1016/j.patrec.2011.01.005.

SUNDARAKRISHNAN NARAYANAN (Student Member, IEEE) is currently pursuing the degree in Computer Science at R. V. College of Engineering, Bengaluru, with a keen interest toward web development and machine learning.

He is also serving as the Webmaster for IEEE-RVCE Student Branch for the year 2024. During this tenure, he also led the development of GateKeeper an automated online approval system used at IEEE-RVCE for event permissions, saving a lot of time and efforts. His research interest includes machine learning, specifically computer vision, demonstrated by his previous work in the domain of road traffic safety. He is keen on developing innovative solutions in the field of road safety and efficient road transport with a lot of work in the same domain. He secured first place in the Bengaluru Mobility Challenge 2024 and was also a Finalist in the IEEE YESIST12 2024 held in Tunisia.

SOHAN VARIER is currently pursuing the degree in Computer Science at R. V. College of Engineering, Bengaluru, with a keen interest toward machine learning, data science, and statistics. His previous projects also revolved around the realm of computer vision, where he tried the 3D reconstruction from 2D images. His research interests include building new and innovative solutions in the fields of software and IoT applications.

TARUN BHUPATHI (Student Member, IEEE) is currently pursuing the degree in Computer Science at R. V. College of Engineering, Bengaluru, specializing in the fields of data science, computer vision, and machine learning. His current research interest includes developing models to improve test analysis within the realm of natural language processing. His research interests include applying machine learning techniques to tackle critical challenges and develop sustainable solutions. In addition to his research, he aims to provide solutions for real world issues using data-driven approaches. He also has researched the integration of machine learning-based systems for congestion detection in computer networks. His work in the field of road safety and efficient road transport aims to optimize traffic flow and reduce congestion in urban areas using data from camera feeds.



MANASWINI SIMHADRI KAVALI is currently pursuing the degree in Computer Science at R. V. College of Engineering, Bengaluru. Previously, she has published her work in the domain of machine learning, reflecting her commitment to contributing valuable insights to the academic community. Her research interests include machine learning, quantitative finance, and computer vision. This biography accompanies her second publication, in which she delves into innovative solutions for road safety and efficient road transport, a field she is passionate about. Her work aims to address pressing challenges in transportation by leveraging her knowledge in machine learning and computer vision to develop applications that have an impact on the society.

MOHANA is currently an Assistant Professor with the Computer Science and Engineering (Cybersecurity) Department, RVCE. He has taught courses on cybersecurity, computer vision, deep learning, and artificial intelligence. He has published more than 100 research articles. His research interests include cyber security, deep learning and AI, quantum computing, and computer vision.

P. RAMAKANTH KUMAR (Senior Member, IEEE) is currently a Professor and the Head of the Department of Computer Science and Engineering, RVCE. He has taught courses on network programming and cybersecurity for industry 4.0. He has published more than 100 research articles. He has executed several funded research and consultancy projects sponsored by DRDO, ISRO, CAIR, LRDE, AICTE, GE India Pvt. Ltd., CABS, and HPE. His research interests include digital image processing, pattern recognition, and natural language processing.

K. SREELAKSHMI (Senior Member, IEEE) is currently a Professor and the Head of the Department of Electronics and Telecommunication Engineering, RVCE. She has taught courses on microwave communication, nano electronic devices, and artificial intelligence. She has executed several funded research and consultancy projects sponsored by AICTE and Samsung. She has published more than 70 research articles. Her research interests include microwave communication and AI.