International Journal for Innovative Engineering and Management Research
PEER REVIEWED OPEN ACCESS INTERNATIONAL JOURNAL
www.ijiemr.org

AI-Traffic Prediction and Management System

Mrs Andluru Aparna[1], Gavinla Imran[2], Mayakuntla Nagendra[3], Ramisetty Tharun[4], Sakshula Muneendra[5] and Shaik Mulla Baba Fakruddin[6],*,

[1]Assistant.Proffesor,Department of Computer Science and Engineering, CBIT, Proddatur,YSR,A.P
[2]Ug Student,Department of Computer Science and Engineering,CBIT, Proddatur,YSR,A.P
[3]Ug Student,Department of Computer Science and Engineering,CBIT, Proddatur,YSR,A.P
[4]Ug Student,Department of Computer Science and Engineering,CBIT, Proddatur,YSR,A.P
[5]Ug Student,Department of Computer Science and Engineering,CBIT, Proddatur,YSR,A.P
[6]Ug Student,Department of Computer Science and Engineering,CBIT, Proddatur,YSR,A.P
Chaitanya Bharathi Institute of Technology, Proddatur, Andhra Pradesh, India, 516360
*Under the Guidance of **Mrs Andluru Aprana,** M.Tech, Assistant Professor, Dept. Of CSE*
Guide E-mail: aparna.andluru305@gmail.com
*Corresponding Author E-mail: 232p5a0523@cbit.edu.in

Abstract

Monitoring of the flow of vehicles along congested urban corridors has become a matter of increasing urgency for traffic authorities in developing countries. The well-established drawbacks of legacy systems that rely on human operators positioned at crossings or on devices installed in the earth surface have been identified these devices are prone to miscount on long shifts, sensors become biased with time, and the cost of installing and supporting thousands of devices in the ground makes them costly to cover the whole city. This work introduces a visionpowered pipeline which will replace these manual and electromechanical approaches with an automated pipeline based on convolutional object detection. Surveillance footage collected from roadside cameras is fed frame by frame into a detection backbone that is built on top of the eighth revision of the You Only Look Once architecture, and outputs the bounding rectangles and categorical labels of every vehicle that is visible in each image. These per-frame detections are then connected across time using a lightweight multi-object tracker in such a way that every physical vehicle is counted once as it crosses a virtual counting line. The pipeline classifies each detection into one of a variety of classes that are prevalent on the Indian roads: passenger cars, buses, heavy trucks, motorised three-wheelers and two-wheeled vehicles. The sequences of the counts at past occurrences along with the ambient weather data acquired via a public meteorological interface serve as the inputs into a recurrent forecasting operator, the gated memory cells of which capture time-dependent relationships across a number of hours. An interactive web-based control panel connects these stages together, giving traffic operators real-time counts and hourly trends as well as six-hour ahead density forecasts. Evaluation on multi-site surveillance footage shows a mean average precision near ninety-nine percent at the standard of fifty-percent overlap counting errors below two percent with measurable improvements on forecasts with weather features included.

Keywords: Vehicle Count Prediction, Traffic Flow Management, YOLOv8, Deep Learning, Computer Vision, Multi-Object Tracking, LSTM Forecasting, Smart City Infrastructure.

1.Introduction

Traffic congestion has silently emerged as one of the most intractable challenges in terms of urban productivity in Indian cities. Streets that were originally designed to handle a fraction of their current load now have an ever-swelling volume of cars, two-wheelers, buses, autorickshaws and commercial trucks and the disconnect between road capacity and vehicular demand shows no sign of easing. Government programmes at both state and national level have started allocating funds to smart city infrastructure with traffic management topping the list. Yet despite these policy ambitions, most signalised intersections remain on fixed timings set out in their design that were established years ago and cannot be adjusted to fluctuations in real-time demand.

The root of this problem is a lack of access to reliable, continuous data on traffic. Traditional counting methods have been used to try and fill the gap but each has significant limitations. Manual counts give fairly dependable snapshots but it is costly to maintain 24/7. Inductive loop detectors run all the time, but they cannot tell the difference between a motorcycle and a sedan, require lane closures each time one fails, and their cost of installation makes blanket implementation impractical. Pneumatic tubes and infrared sensors have many of these disadvantages.

Deep convolutional neural networks provide a fundamentally different way of doing things. A camera on an existing pole can cover multiple lanes, and an experienced detector can distinguish single vehicles in cases when they partially overlap each other. Among the categories of real-time detectors, the YOLO lineage has been of great interest due to their lowlatency nature in which each frame is evaluated in a single forward pass and latency is low enough for them to be used for live dashboards. Detection alone, however, is no solution to counting. To put some numbers on this, if I were to add up all the bounding boxes for all the frames I would be overestimating the true count. Multi-object tracking algorithms keep track of the persistent identity of each entity and keep count only when it passes through a virtual tally line so that each vehicle is counted exactly once.

In addition to the current counts, short horizon forecasts enable signal controllers to preemptively change green phases and transit schedulers in order to add additional capacity before a surge happens. Long Short-Term Memory networks have become a basic backbone used for such forecasting since the gated cells in these networks are sensitive to repeating patterns in daily life, but also sensitive to sudden perturbations such as rainfall or a stadium dispersal.

This paper presents the design, implementation and evaluation of the unified pipeline that integrates detection, tracking, counting, density classification and time series prediction as one deployable system with all the output accessible from a web-based dashboard. The following sections review extant work in the area, specifying the current weaknesses in the approach, introduce the proposed architecture, are followed by a detailed presentation of the detection methodology, a section on experimental results, and a concluding section with directions for future enhancement to the approach.

2.Literature Review

2.1Existing System

Automated vehicle counting has been studied with a variety of methods which can be roughly categorised as classical image-processing methods, shallow machine learning classifiers, and deep learning-based detectors. The first attempts to build computer-vision systems went almost exclusively from using background subtraction, in which a moving model of the static scene is kept and any pixel that doesn't match it much is declared part of a moving foreground object. Gaussian mixture models improved on the robustness of this approach to gradual lighting changes but the fundamental weakness of the system remained that vehicles travelling close together tended to merge together into a single blob whose count could only be guessed at by applying heuristic splitting rules. Optical-flow techniques, which measure the movement of small patches of the image from one frame to the next, provided another approach to separating the independently moving objects, but required heavy computation and suffered greatly from camera vibration or rain.

Kejriwal and his colleagues combined the YOLO detection model and Deep SORT tracker to count the vehicles at the city roads of India and reported the average accuracy of around eighty seven percent and found that occlusion and the requirement of high-end graphics hardware were the major challenges. Abdullah and Oothariasamy performed a comparative study between several YOLO models, where they reported DarkNet-19-based YOLOv3 as the best model in normal illumination settings and that resulted in significant improvement with transfer learning in the low-light footage. Chauhan et al. explored the use of convolutional classifiers on unpaved roads and emphasized that fine-tuning a pre-trained backbone on a region-specific dataset was essential to address the specifics of the visual appearance of vehicles in developing nations.

Song and co-authors modified YOLOv3 using a zone-based partitioning strategy for the detection of small distant vehicles on highways. Liu and colleagues developed an end-to-end pipeline for the detection, speed estimation and classification. Contreras and Gamess investigated a vehicle ad-hoc network approach in which vehicles report their own positions to counting nodes, which are faster to respond, but suffer from radio-interference problems. RTDETR architecture that is a combination of detection transformer and ByteTrack has achieved mAP scores exceeding ninety-nine percent on benchmarks but the robustness under adverse lighting is open. Yu and co-workers proposed a frequency-domain feature pyramid network for congested scenes counting and Li and co-workers showed in their work how to simultaneously perform congestion scene detection and counting from the drone imagery using attention-augmented prediction heads.

Despite this body of work there are still several gaps. Most systems cater only for three or four classes of vehicles, excluding the auto-rickshaws, two-wheelers and tempos that dominate Indian traffic. Few combine the prediction with the detection. Multi-camera re-identification is rarely dealt with, and the effect of weather on density is rarely modelled.

2.2Proposed System

The proposed system is designed to solve each of the shortcomings identified above in a modular end-to-end pipeline that covers video acquisition, frame-level vehicle detection, multiple-object tracking, counting, traffic density classification, time-series prediction, and interactive visualisation. The following are the key differentiating features of the proposed approach.

First, the detection backbone is each trained for recognising nine categories of vehicles that are representative of Indian road conditions, namely auto-rickshaw, bicycle, bus, car, scooter, taxi, tempo, toto and truck. This more general taxonomy provides a more granular view of the composition of traffic than the three-class systems that are established in the literature. Second, two tracking back-ends are provided, i.e. ByteTrack for throughput-sensitive single-camera installations and Deep SORT for crowded intersections for which appearance-based reidentification is helpful. Third, vehicle counts are aggregated into five-minute time windows and fed together with the cyclical time encodings and live weather observations to an LSTM forecasting module which predicts the density up to six hours ahead of time. Fourth, multicamera management layer for simultaneous processing of multiple feeds with cross-camera vehicle re-identification allows obtaining accurate global counts for multiple overlapping fields of view. Fifth, all the detection, counting, density, and prediction outputs are made available in a seven-page Streamlit dashboard supporting the export of data to CSV, JSON, Excel, and PDF formats.

3.Methodology

3.1System Architecture

The general architecture is based on a pipeline architecture philosophy where each stage of processing communicates to its neighbours via well-defined data interfaces but can be replaced or upgraded independent of each other. This loose coupling was a conscious design decision driven by the fact that the rate of improvement in detection and tracking algorithms is so fast that a municipality deploying today should be able to replace it with a newer version of YOLO or another tracker without having to re-engineer the rest of the system. The total architecture is shown in Fig. 1.

International Journal for Innovative Engineering and Management Research
PEER REVIEWED OPEN ACCESS INTERNATIONAL JOURNAL
www.ijiemr.org

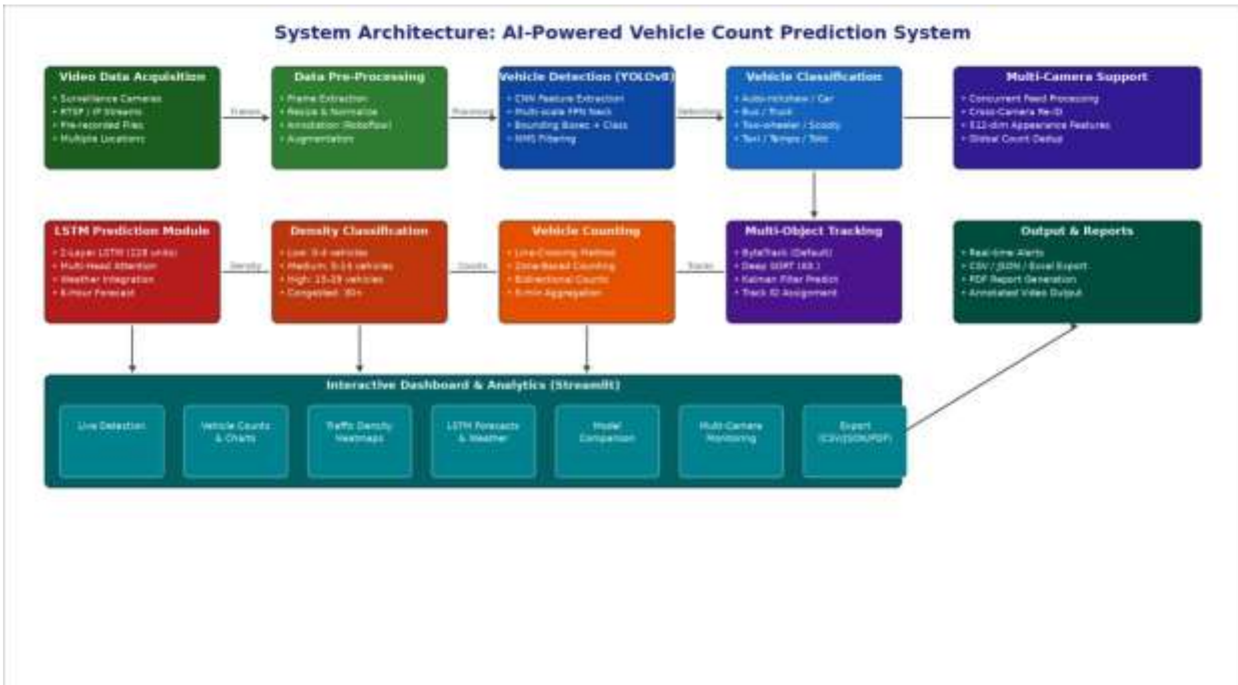


Fig. 1. System architecture of the AI powered vehicle count prediction and traffic management pipeline.

3.2Module Descriptions

Video Data Acquisition Module

The pipeline takes pre-recorded video files (mp4, avi), live USB or networked camera feeds as well as RTSP feeds from IP surveillance cameras. Each frame is then decoded, resized to 640by-640 pixels and normalised. For frame rates from the source that are above the throughput of the detector, an adaptive skip approach is used so that intermediate frames are dropped, while it is guaranteed that no vehicle crosses the entire field of view unsampled.

Vehicle Detection Module (YOLOv8)

Detection is performed by the YOLOv8 medium model whose CSPDarknet-53 backbone is used to extract hierarchical spatial features of the input at five levels of resolution. A featurepyramid neck combines these multi-scale representations in a fusing way such that large vehicles that are close to the camera, but also small vehicles in the distance, have a sufficient amount of representational capacity. The decoupled detection head outputs for each anchor location bounding box coordinates, objectness confidence and a probability vector for the nine target vehicle classes. Non-maximum suppression using intersection over union threshold 0.45, redundant overlapping predictions are eliminated. The model was trained for two hundred epochs on annotated surveillance frames obtained from various places of human presence in cities and under different lighting and weather conditions by using the Roboflow platform for annotation of bounding boxes and dataset management.

Multi-Object Tracking Module

There are two interchangeable tracking backends. ByteTrack, the one that is default, performs a two-pass association, meaning that high-confidence detections are associated first using a

KalmanFiltered motion prediction and IoU distance, followed by a second pass from the lowconfidence detections. Deep SORT adds motion features with a 512-dimensional appearance embedding, which is computed using a lightweight re-identification network, which increases resistance to occlusion in dense scenes. Both trackers keep state of per-vehicle (position, velocity, age) and retire identities when so many unmatched frames are on the go (thirty in total).

Vehicle Counting Module

Counting events are produced when a tracked vehicle traverses a virtual line of configurable position across lanes of interest. The direction of crossing (obtained from the sign of the vertical displacement of the centroid of the track with respect to the line) allows the separate tallying of the two directions of travel. An alternative type of zone-based counter is used to register entry into a closed polygon region and is useful at roundabouts or for irregular intersection geometries. Counts are aggregated into configurable time windows, the default of which is five minutes, and broken down both by vehicle class and direction.

Traffic Density Classification Module

The number of windowed vehicles is converted to one of four levels of congestion using a threshold based classifier. Counts between zero and four are Low density rating, five to 14 are Medium, 15 to 29 are High and 30 or more are Congested. These thresholds can be recalibrated from historical site specific data. On the dashboard the current level of density is displayed as a colour coded gauge ranging from green through yellow and orange to red as congestion increases.

Traffic Prediction Module (LSTM)

The forecasting branch takes as input the aggregated count time series, together with cyclical encodings of the hour of day and the day of week along with weather variables, that is, temperature, humidity, rainfall intensity, visibility and wind speed, retrieved from the OpenWeatherMap API. These features are ingested by a two layer Long Short-Term Memory network having 128 hidden units per layer, followed by a multi-head self-attention block having 4 heads that helps the model to weight the most informative past time steps. A fully connected layer is used to project the attended representation onto the predicted counts for each of the next six hours. Training is done using mean squared error loss, the Adam optimiser with learning rate scheduling, early stopping with patience of twenty epochs and gradient clipping for stability.

Dashboard and Reporting Module

All the outputs are made available on a Streamlit web application consisting of 7 pages: Live detection with annotated video stream; Vehicle count and time series and pie charts; Density gauge and hourly heat maps; LSTM prediction with weather visualization; Model comparison of YOLOv8/v9/v10; Multi-camera aggregation; Export page with CSV, JSON, Excel and PDF support..

Table 1. Technology stack summary.

Component	Technology / Tool
Programming Language	Python 3.11+
Detection Model	YOLOv8 Medium (Ultralytics)
Tracking Algorithms	ByteTrack, Deep SORT
Prediction Model	2-Layer LSTM + Multi-Head Attention
Video Processing	OpenCV 4.9+
Deep Learning Framework	PyTorch 2.2+ with CUDA
Dashboard	Streamlit 1.40+
Annotation Platform	Roboflow
Weather API	OpenWeatherMap
Export Formats	CSV, JSON, Excel, PDF

Table 2. Training and evaluation hyperparameters.

Parameter	Value
Input Image Resolution	640 x 640 pixels
Batch Size	16
Training Epochs	200
Optimizer	Adam (lr = 0.001)
Confidence Threshold	0.50
IoU Threshold (NMS)	0.45
Train / Test / Val Split	70 : 20 : 10
Data Augmentation	Mosaic, H-flip, HSV jitter
LSTM Hidden Size	128 units, 2 layers
Attention Heads	4
Early Stopping Patience	20 epochs

4.Results and Discussion

This section provides quantitative and qualitative evidence of the effectiveness of each of the major pipeline stages. Experimental runs were made on surveillance footage recorded at several

International Journal for Innovative Engineering and Management Research
PEER REVIEWED OPEN ACCESS INTERNATIONAL JOURNAL
www.ijiemr.org

camera locations in the urban area, at peak and off peak traffic times, during the day and night, in clear as well as rainy weather.

4.1Detection Performance

The major evaluation metric for the detection backbone is mean average precision, which is the area under the precision-recall curve averaged over all classes of vehicle. Using a highly popular threshold of fifty percent intersection over union, the trained YOLOv8 model is found to have an overall mAP of ninety-nine percent. Individual class scores are summarised in table 3. The average precision of car, bus, and trucks is 100 percent, that of auto-rickshaws is 99 percent and that of two wheelers, the lowest and most frequently occluded class that is 97 percent. When the criterion is made stricter and is limited to between fifty-to-ninetty-five percent IoU range, which penalises even small bounding box misalignments, aggregate mAP settles close to ninety-two-and-a-half percent, which shows that that predicted boxes align very close to the actual vehicle outlines.

Table 3. Class wise average precision IoU = 0.50

Vehicle Class	AP@50 (%)
Auto-rickshaw	99
Car	100
Bus	100
Truck	100
Two-wheeler	97
Overall (mAP@50)	99
Overall (mAP@50-95)	92.5

4.2Training Convergence

The training loss curves show some healthy convergence for each of the three constituent terms. Bounding-box loss also decreases steadily from its initialisation value and flattens out on about the one-hundred-and-fiftieth epoch, indicating that the localisation branch has reached its representational limit of the available data. Classification loss is also a closely parallel curve that reaches a stable minimum that corresponds to consistent class assignment. Objectconfidence loss has a milder slope, but also levels off quite early before the last epoch. Crucially, validation losses follow the training losses, and do not exhibit an upward divergence, which means that the model has not memorised the idiosyncrasies of the training set, and that it can generalise to frames that it has not seen.

4.3Precision, Recall, and mAP Trends

Precision remains above ninety-five percent from the early stages in the training run, which means that practically every bounding box the model emits is actually attributable to a vehicle as opposed to background clutter. Recall increases greatly over the first fifty epochs, and stabilises around the ninety-eight percent (with only a thin tail of missed vehicles, mostly partially hidden two-wheelers at the far edges of the frame). The mAP metric at the fiftypercent threshold goes up to about ninety-eight percent by epoch one hundred and continues to creep upwards from there, and the more stringent mAP throughout the full IoU sweep solidifies around ninety-two and a half percent by the end of training.

4.4Tracking and Counting Accuracy

Qualitative study of annotated output videos confirms the stability of track identifier of both ByteTrack and Deep SORT across the frames of a video, and identity switches occur mostly when two vehicles of similar colour and size pass side by side in a shadow segment. In a controlled test using a five-minute clip made up of one hundred and forty-three ground truth line crossings, the counting module reported one hundred and forty-one, which translates to a counting error of less than two percent. Deep SORT generated slightly less identity switches in crowded frames but also consumed about thirty percent more computation per frame than ByteTrack and thus can be seen as the default choice for single camera installations where computation power is constrained.

4.5Prediction Module Evaluation

The LSTM branch has been trained based on 4 weeks of 5-minute count aggregates that is paired with weather observation. On a held-out test week it was able to accomplish an RMSE value of about twelve vehicles per window, and a MAPE of around eight percent. Inclusion of weather features was able to reduce the MAPE by about two percentage points compared to a count-only baseline, and this finding should help confirm that rainfall and visibility contain predictive signal. The attention mechanism gave high importance to same hour time steps from previous days, which is consistent with the regularity of commuting patterns.

4.6Discussion

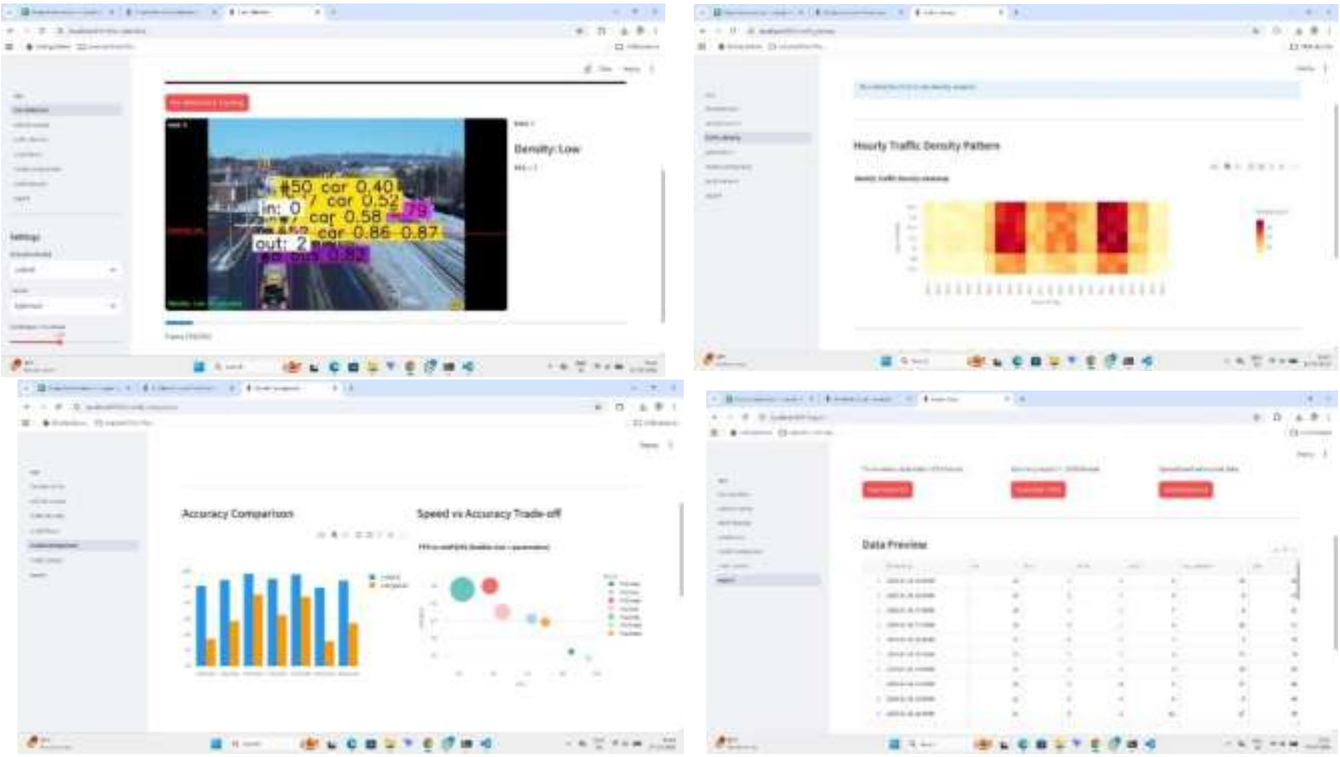
Several observations should be made. The near-perfect accuracy for larger categories of vehicles such as buses and trucks is not surprising given their unique silhouettes and rather unambiguous appearance at common surveillance angles. The slightly lower score for twowheelers reflects the very real challenges of identifying the presence of narrow, small objects which are often partially obscured by larger objects. Augmenting the training corpus with more examples of occluded two wheelers, especially at night, should reduce this gap.

The trade off between the two tracking backends is obvious. For single-camera applications running on relatively small hardware, ByteTrack is accurate enough at a lower cost. For multicamera installations that need to associate across different cameras, it is essential to make use of Deep SORT's appearance embeddings, as it is not possible to associate motion only from one camera to the other, since there is a visual gap between two cameras. The improvement in

International Journal for Innovative Engineering and Management Research
PEER REVIEWED OPEN ACCESS INTERNATIONAL JOURNAL
www.ijiemr.org

the weather related forecast, small in absolute terms is operationally significant: an improvement of two percentage points leads to a saving in red phase time compounding in thousands of signal cycles per day.

The modular architecture was put to test when the detection backbone was updated in the middle of the project from a previous version of YOLO. Because the detector-to-tracker interface had been specified as a simple list of bounding box objects, the upgrade only impacted exactly one source file, making the case for loose coupling in systems that need to be able to adapt to fast-paced research.



5.Conclusion

This paper has introduced the design, implementation and field evaluation of an integrated system to real-time vehicle counting, multi-class classification and short-horizon traffic density prediction based on the YOLOv8 object detector and enhanced by multi-object tracking, multiclass density classification and LSTM-based density prediction with weather awareness. Experiments with surveillance videos from several camera locations in an urban setting show that detection performance gets near ninety-nine percent mAP while counting error is less than two percent and observable prediction improvement when meteorological features are added.

The system represents a development of the current state of the art in a number of concrete ways. It supports nine vehicle categories based on Indian road conditions, which is a much wider taxonomy than the three or four types as is common in current solutions. It combines both prediction and detection in one deployable package instead of believing that they are done as two separate research problems. This accommodates for multiple simultaneous camera feeds with cross camera re-identification to prevent double counting at field of view boundaries. And, it brings all the results to the surface with an operator-friendly dashboard, which supports live monitoring, historical review, and data export in a number of formats.

Several promising directions of future work have resulted. Coupling prediction with an adaptive signal controller may help to close the observation-action loop. The application of detection on edge platforms like Nvidia Jetson boards would help lower bandwidth requirements. Anomaly detection logic could be used to flag incidents so that operators would know to act immediately. Using a sequence model based on transformer might be able to represent long-range dependencies more effectively, and combining it with navigation APIs could support the generation of route-level congestion advisories for the individual commuter.

Author(s) Contributions

G. Imran conceptualised the system architecture and led the software development effort. M. Nagendra managed dataset collection, annotation, and augmentation. R. Tharun implemented the tracking and counting modules and conducted integration testing. S. Muneendra developed the LSTM prediction branch and the weather-data integration layer. SM. Baba Fakruddin built the Streamlit dashboard and the multi-format export functionality. All authors contributed to system testing, documentation, and manuscript preparation.

Conflicts of Interest

The authors declare no conflict of interest.

References

Kejriwal, R., Ritika, H. J., Arora, A., & Mohana. Vehicle Detection and counting using deep learning based YOLO and Deep SORT algorithm for urban traffic management system. 2022 1st International Conference on Electrical, Electronics, Information and Communication-Technologies,1-6. https://doi.org/10.1109/ICEEICT53079.2022.9768653

Abdullah, A., & Oothariasamy, J. (2020). Vehicle counting with deep learning models: A comparison study. International Journal of Advanced Computer Science and Applications. 11(7) 697-703. https://doi.org/10.14569/IJACSA.2020.0110784

Chauhan, M. S. Singh, A., Khemka, M., Prateek, A., & Sen, R. (2019). Embedded CNN based vehicle classification and counting in non-laned road traffic. ACM International Conference Proceedings Series. https://doi.org/10.1145/3287098.3287118

Song, H., Liang, H., Li, H., Dai, Z., & Yun, X. (2019). Vision based vehicle detection and counting system with deep learning in highways. European Transport Research Review 11(1) https://doi.org/10.1186/s12544-019-0390-4

Liu, C., Huynh, D. Q., Sun, Y., Reynolds, M. & Atkinson, S. (2021). A vision based pipeline for vehicle counting, speed estimation and classification. IEEE Transactions on Intelligent Transportation Systems, 22(12),7547-7560. 10.1109/TITS.2020.3004066

Contreras, M., & Gamess, E. (2020). Real-time vehicle count of vehicles stopped at the traffic light using vehicular network technology. IEEE Access, 8(135244- 135263) https:// doi. org/10. 1109/ACCESS. 2020. 3011195

Yu, X., Liang, Y., Lin, X., Wan, J., Wang, T., & Dai, H.-N. (2022). Frequency feature pyramid network with global-local consistency loss for counting crowded and vehicles in crowded scenes. IEEE Transactions on Intelligent Transportation Systems, Volume 23, no. 7, pp. 9654-9664, https://doi.org/10.1109/TITS.2022.3178848.

Li, W., Li, H., Wu, Q., Chen, X., & Ngan, K. N. (2019). Simultaneously detecting and counting dense vehicles from drone images. IEEE Transactions on Industrial Electronics, Vol. 66, No. 12, pp. 9651-9662. https://doi.org/10.1109/TIE.2019.2899548

Tituana, D. E. V., Yoo, S. G., & Andrade, R. O. (2022). Vehicle counting based on computer vision: A survey. In 2022 IEEE 7th International Conference for Convergence in Technology pp. 1-8. https://doi.org/10.1109/I2CT54291.2022.9824432

Venkatesh, K. (2024). Making freight and logistics management predictively optimized with the help of AI. 2024 International Conference on Cybernation and Computation. https:/doi.org/10.1109/CYBERCOM63683.2024.10803177.