AutoTrack: A Deep Learning approach on a Novel Dataset for Vehicle Detection and Route Tracking from CCTV Images

***Abstract*—Hit-and-run incidents remain a critical public safety challenge, with thousands of fatalities reported annually. In Bangladesh alone, 7,902 road accident deaths occurred in 2023, with motorcycles involved in 32.4% of cases. Traditional investigative approaches, such as manual CCTV analysis, are inefficient due to poor camera quality, sparse coverage, and human error, leading to significant delays in offender identification. This paper presents an automated system integrating YOLOv5 for vehicle color-based detection and YOLOv8 for license plate recognition to enhance hit-and-run investigations. The proposed system reconstructs vehicle paths using timestamped footage, resolves identification ambiguities, and provides real-time geospatial mapping for law enforcement intervention. Preliminary evaluations demonstrate an 83% accuracy rate in tracking vehicles across multiple surveillance feeds, reducing investigation time from hours to minutes. While challenges such as infrastructure costs and inconsistent CCTV coverage persist, our approach offers a scalable solution for improving road safety and justice delivery. By automating vehicle tracking and identification, this system strengthens law enforcement capabilities, increasing accountability and reducing hit-and-run incidents.**

***Index Terms*—Detection, Vehicle Tracking, License Plate Recognition (LPR), Automated Surveillance, Geospatial Mapping, Traffic Incident Analysis, Law Enforcement Technology.**

I. Introduction

The rapid development of smart cities has exposed critical gaps in addressing hit-and-run incidents, a global public safety crisis claiming thousands of lives annually. In Bangladesh alone, 7,902 fatalities occurred in road accidents in 2023, with motorcycles implicated in 32.4% of cases. Tragedies like the May 2024 Dhaka-Aricha highway incident, where a police officer's wife was killed while the offender escaped, highlight systemic failures in the delivery of justice [1].

Hit-and-run incidents leave victims and their families in a state of distress, often compounded by the inefficiencies of traditional investigative methods. Victims, despite their best efforts to recall critical details such as vehicle color or license plate numbers, face significant barriers in achieving justice. Law enforcement agencies, in turn, rely on manual CCTV footage analysis, a process fraught with inefficiencies. Poor camera quality, sparse coverage, and labor-intensive review processes often delay investigations, allowing offenders to evade accountability. Studies indicate that manual reviews can take between 12 to 48 hours, providing ample time for suspects to alter their vehicles or flee jurisdictions [2].

From the perspective of law enforcement, hit-and-run investigations present a daunting challenge. Officers must first gather eyewitness accounts and then sift through hours of surveillance footage, often of subpar quality. In Dhaka, for instance, 43% of CCTV footage is unusable due to low resolution or inadequate lighting. This reliance on manual processes introduces significant delays and increases the likelihood of human error, with studies showing that officers overlook critical frames in 22% of cases during exhaustive footage scans. Without automated tools, investigations depend on chance whether the right camera was operational or whether it captured the vehicle's license plate.

Current systems have three major issues: data limitations, where 43% of CCTV footage in Dhaka is unusable due to low resolution or poor lighting; time sensitivity, as manual reviews take 68% longer than automated systems; and human error, with officers missing critical frames in 22% of cases [3].

While ideal investigations require high-quality cameras and instant footage access, reality demands pragmatism. We propose an automated system combining YOLOv5 (color-based detection) and YOLOv8 (license plate recognition) to track vehicles across CCTV networks. Unlike existing tools, our approach:

- **Reconstructs Paths:** Analyzes timestamped footage to map suspect routes.
- **Resolves Ambiguities:** Flags multiple vehicles matching descriptions (e.g., white SUVs) until plate data confirms identity.
- **Guides Law Enforcement:** Provides optimal interception routes via geospatial mapping.

Each year, 20% of global road deaths involve hit-and-runs. By automating detection and path reconstruction, our system empowers law enforcement to act decisively—turning victims' despair into hope and making roads safer for all [4].

The whole paper is organized in the following order where section II provides the Literature Review, section III provides Proposed System, section IV provides Experimental Results and section V provides Conclusion for our system.

II. LITERATURE REVIEW

In a recent study, Ning Chen et al. [5] proposed an AI-driven vehicle tracking system that leverages deep learning techniques for license plate recognition and anomaly detection in intelligent transportation systems. This method has been designed to improve the identification of suspect vehicles, aiding in crime prevention and traffic management. In similar fashion, Walid Balid et al. [6] have introduced a multi-camera tracking framework capable of reconstructing vehicle paths across multiple surveillance feeds. Their system has been demonstrated in handling occlusions and complex urban traffic scenarios.

Alternatively, Attiq ur Rehman et al. [7] have focused on the integration of image processing and sensor fusion for vehicle detection and classification. Their study has highlighted the challenges of low-light conditions and motion blur, proposing an adaptive filtering technique to enhance recognition accuracy. Moreover, Xi Zhao et al. [8] have proposed a crowdsourcing-based vehicle profiling framework, where users can submit real-time incident reports, assisting in hit-and-run case investigations. However, their system has relied heavily on manual inputs, limiting its efficiency in real-time tracking.

Benjamin Coifman et al. [9] have explored vehicle movement prediction using spatiotemporal analysis, employing GPS and surveillance footage to track vehicles' last known locations. Their study has demonstrated how timestamped data can be used to refine trajectory estimations, which aligns closely with our approach. In addition, Bin Tian et al. [10] have examined path reconstruction using a combination of surveillance cameras and AI-based motion prediction models. Their work has addressed the challenges of multi-vehicle interactions, proposing a filtering mechanism to differentiate between similar vehicles at intersections.

Furthermore, Chao-Yung Hsu et al. [11] have developed a high-performance license plate recognition system using a hybrid deep learning approach. Their model has combined CNN-based feature extraction with an OCR engine for high-accuracy character recognition. The study has shown that integrating pre-trained OCR solutions can significantly enhance processing speeds while maintaining detection precision.

While these approaches offer valuable insights, they lack **a fully automated solution for real-time hit-and-run case tracking**, which is the core focus of our work. Most existing research either relies on **manual data collection** (e.g., crowdsourcing or law enforcement intervention) or focuses on **static vehicle detection** without constructing a **continuous path of movement.**

Our Contribution and Enhancement:

- **Automated Multi-Camera Tracking:** Unlike existing methods that track vehicles in isolated CCTV footage, our system continuously follows a vehicle's movement across multiple cameras, reconstructing its full trajectory.
- **Dynamic Suspect Filtering:** Our approach refines suspect identification dynamically, handling multiple vehicles of similar appearance at intersections by leveraging timestamped data and sequential tracking.
- **Path Generation with Real-Time Updates:** We integrate timestamp-based metadata and use **Google Maps API** to display the **last known position** of the suspect vehicle, optimizing the search for law enforcement.
- **Minimizing Manual Effort:** We automate the vehicle tracking process using **deep learning models** for detection, **OCR for license plate extraction**, and **trajectory mapping**, significantly reducing the need for manual intervention and improving response time.

From our review, we see that while significant progress has been made in vehicle detection, tracking, and license plate recognition, gaps remain in **automated multi-camera tracking, dynamic path reconstruction, and real-time suspect filtering.** Many existing studies have focused on static detection rather than **continuous trajectory mapping**, which is essential for identifying vehicles involved in hit-and-run cases. Our proposed system aims to address these limitations by integrating timestamped vehicle tracking across multiple surveillance cameras and leveraging deep learning for real-time path estimation.

III. PROPOSED SYSTEM

This system integrates three key modules – Vehicle Detection, License Plate Recognition, and Path Generation – to automatically identify, track, and map the routes of vehicles involved in hit-and-run incidents. Designed for real-time CCTV processing, our system aims to deliver rapid, actionable intelligence to law enforcement.

A. Dataset, Preprocessing and Augmentation

*1) **Dataset Overview**:* The dataset for training the vehicle detection model consists of **4856** raw images specifically curated for vehicle detection based on color. The images were sourced from diverse **CCTV** camera feeds under varying environmental conditions, including different lighting conditions, weather patterns, and traffic densities. The dataset was annotated by manually labeling with bounding boxes and their colors using a dedicated annotation tool(**Roboflow**). To mitigate class imbalance and enhance model robustness, especially for underrepresented vehicle colors (**maroon, green, red**), saturation augmentation techniques were employed to artificially increase the representation of these classes.The dataset was partitioned into training, validation, and testing sets to ensure rigorous model evaluation and prevent overfitting.

*2) **Image Preprocessing and Augmentation**:* To standardize input data and improve model generalization, the following preprocessing steps were applied:

- **Resizing:** All images were resized to **640 × 640 pixels** to ensure consistent input dimensions for the deep learning models.
- **Horizontal Flipping:** In horizontal flipping, images are mirrored along the vertical axis to simulate different vehicle viewpoints and directions.

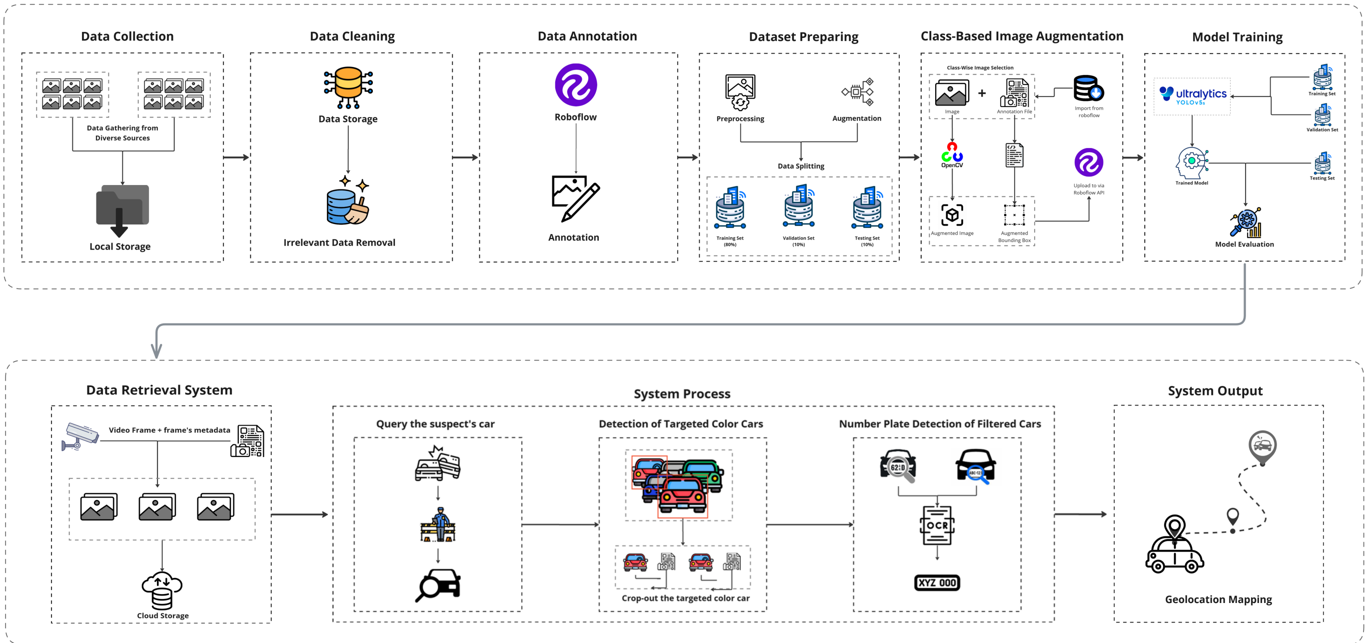

Fig. 1. System Overview

- **Rotation:** In rotation-based augmentation, we adjusted image orientations by rotating them randomly in the dataset. This simulates real-world camera angle variations while preventing excessive distortions.
- **Brightness Adjustment:** Brightness adjustment modifies image intensity to simulate lighting variations from day, night, and weather changes.

B. System Integration and Workflow

This deep learning system automates vehicle tracking for post-accident investigations using urban CCTV networks. It integrates three key models: Vehicle Detection, which filters CCTV footage to identify color-specific vehicles; License Plate Recognition, leveraging YOLOv8 and EasyOCR for accurate plate identification; and Path Generation, which reconstructs vehicle trajectories across multiple cameras using timestamped data. This end-to-end pipeline enables efficient vehicle identification and route mapping, assisting law enforcement in post-accident analysis.

- **Vehicle Detection Model**
  For robust and real-time vehicle detection, we employed the YOLOv5s architecture, a member of the You Only Look Once (YOLO) family of object detection models. YOLOv5 was selected for its advantageous balance between detection accuracy and computational efficiency, making it particularly well-suited for processing high-volume video streams from urban CCTV networks. The 's' variant (YOLOv5s, small) was specifically chosen for its lightweight nature, facilitating faster inference speeds necessary for real-time applications without significantly compromising detection performance.

- **Model Architecture and Training:**
  The model was trained from scratch using our custom annotated dataset described in Section III.A. The training process was configured with the following key parameters:
  Epochs: Training was conducted for 50 epochs. This epoch count was empirically determined through preliminary experiments, balancing model convergence and computational cost.
  Batch Size: A batch size of 16 was used, maximizing GPU utilization within memory constraints while maintaining stable gradient descent.

- **Output and Color Filtering:** The proposed vehicle detection system incorporates a probabilistic classification approach for color identification, assigning each detected vehicle a probability distribution across predefined color classes: Red, Blue, Black, Green, Maroon, Silver, and White. The class with the highest probability is designated as the predicted color, ensuring an optimal match with the detected vehicle's visual attributes.
  To refine detection outputs, a color-based filtering mechanism is integrated post-detection, enabling targeted searches based on specific color criteria. This filtering process ensures that only vehicles matching a queried color (e.g., "red") are retained for analysis, aligning with investigative requirements in forensic and surveillance applications. The selection of color classes is based on their prevalence in real-world

traffic environments and their significance in incident reporting. By incorporating color-based filtering into the model's output interpretation, the system enhances search efficiency, allowing investigators to rapidly identify vehicles matching witness descriptions or initial reports. This methodology significantly improves the precision and effectiveness of automated vehicle identification in large-scale CCTV monitoring and forensic analysis.

– **License Plate Recognition Model:** The License Plate Recognition module consists of a two-stage process:

  License Plate Detection (YOLOv8): In the first stage, YOLOv8 is used to detect and localize license plates from vehicle images. It offers high accuracy, fast processing, and improved feature extraction compared to earlier versions. The model is trained on a diverse dataset to handle variations in lighting, angles, and environmental conditions. Once detected, a bounding box is drawn around the plate, and the cropped region is passed to the next stage for character recognition [12].

  Optical Character Recognition (EasyOCR): In the second stage, EasyOCR [13] is used to perform Optical Character Recognition (OCR) and extract the alphanumeric characters from the detected license plate region. EasyOCR is capable of handling various font styles, plate formats, and image quality degradations common in real-world CCTV footage.

– **Path Generation Model:** This module is responsible for reconstructing the trajectory of tracked vehicles across multiple cameras. It leverages the timestamped metadata associated with each vehicle detection, including camera ID, location coordinates, and detection time. By correlating these metadata points across different camera feeds, the Path Generation Model constructs a chronological sequence of vehicle locations, effectively mapping the vehicle's movement path. This trajectory visualization aids law enforcement in tracking vehicles of interest through urban environments, even across camera blind spots and intersections. The model is designed to handle potential data noise and inconsistencies in CCTV feeds through temporal and spatial data alignment techniques.

IV. Experimental Results

Model	Accuracy	Precision	Recall	F1 Score
YOLOv5s	83.5%	79%	80%	79%
YOLOv11n	83.9%	78%	78%	79%

TABLE I
PERFORMANCE METRICS OF YOLO MODELS

This section details the outcomes of our experimental evaluations, focusing on the performance of the YOLOv5s and YOLOv11n models. We assessed the models using standard object detection metrics: mean Average Precision (mAP), Precision, Recall, and F1 Score. The results, summarized in Table 1, provide a quantitative analysis of each model's effectiveness.

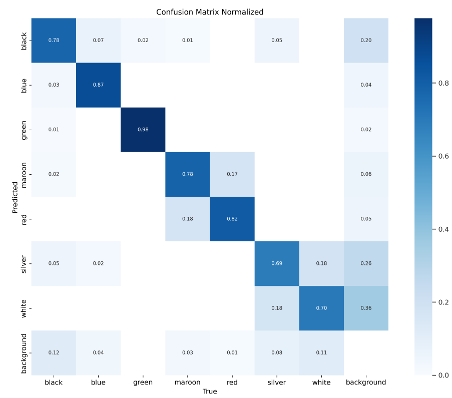

Fig. 2. Normalized Confusion Matrix of YOLOv11.

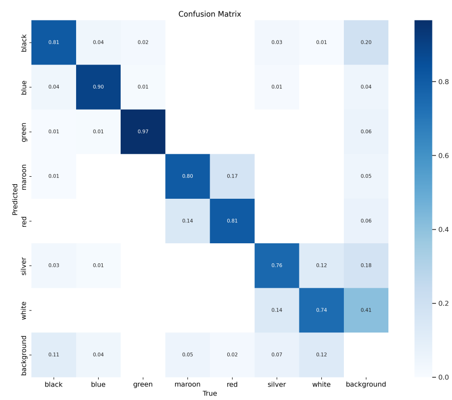

Fig. 3. Normalized Confusion Matrix of YOLOv5s.

Both YOLOv5s and YOLOv11n models, as summarized in Table 1, exhibit strong object detection capabilities. YOLOv11 slightly outperforms YOLOv5s in overall ac-

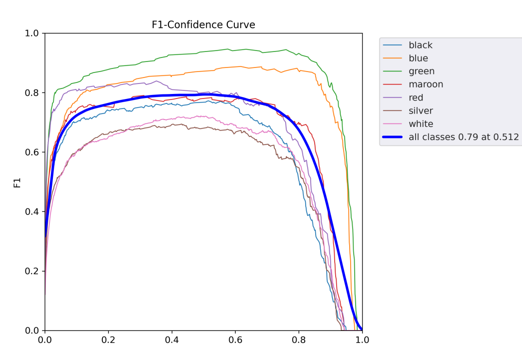

Fig. 4. Illustrate the training performance in terms of recall plotted against the confidence score.

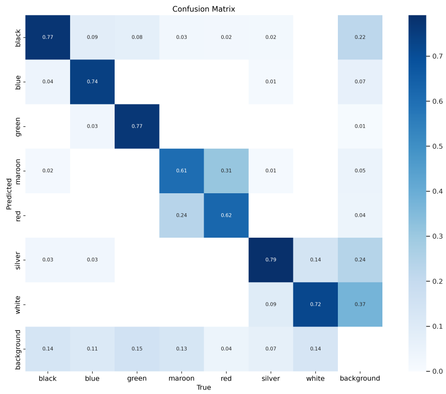

Fig. 5. Normalized confusion matrix of YOLOv5s model, evaluated before preprocessing and data augmentation.

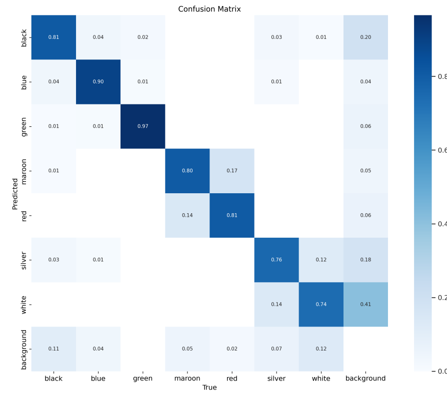

Fig. 6. Normalized confusion matrix of YOLOv5s model, evaluated after preprocessing and data augmentation.

curacy (mAP: 83.9% vs. 83.5%) and Precision (80% vs. 79%), indicating a marginal improvement in minimizing false positives. In contrast, YOLOv5s achieves slightly better Recall (80% vs. 78%), suggesting it might be more effective at capturing all relevant objects, potentially at the cost of some precision. However, with an F1 Score of 79% for both, their overall performance, balancing precision and recall, is essentially equivalent.

Category	Details
Framework / Version	Ultralytics YOLOv5s
Training Details	40 epochs Batch size: 16, Image size: 640x640 Learning rate: 0.01, SGD optimizer with weight decay 0.0005 Data loader workers: 8
Model Architecture	157 layers, 7,029,004 parameters, 15.8 GFLOPS
Model Summary	157 layers, 7,029,004 parameters, 15.8 GFLOPS
Evaluation Metrics	(For Valid sub-dataset) Images: 601, Instances: 1009 Box(P): 0.792, R: 0.799, mAP50: 0.836, mAP: 0.658

TABLE II
YOLOv5s TRAINING CONFIGURATION AND EVALUATION METRICS

A. *Impact of Preprocessing and Data Augmentation:*

To assess the effectiveness of data preprocessing and augmentation techniques, experiments were conducted in two phases:

Before Preprocessing: The initial model was trained on a dataset of 2,216 images, achieving an accuracy of 73.5%, a precision of 71.1%, a recall of 69.6%, and an F1 score of 70.3%. However, the confusion matrix analysis revealed significant misclassification errors, especially among visually similar vehicle colors. Specifically, red and maroon vehicles were frequently confused, as were white and silver vehicles. These misclassifications highlighted dataset biases in color distribution and underscored the model's difficulty in distinguishing between highly similar hues.

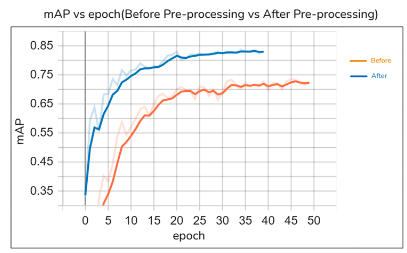

Fig. 7. Illustrates the comparison of mAP before and after preprocessing.

After Preprocessing and Data Augmentation: To improve performance, data augmentation and preprocessing techniques were applied, expanding the dataset to 12,310 images. The augmentation strategies included horizontal and vertical flips, brightness and exposure adjustments, and saturation augmentation, specifically targeting underrepresented colors. As a result, the model's performance significantly improved, achieving an accuracy of 83.5%, a precision of 80.7%, a recall of 79.6%, and an F1 score of 79.5%. This demonstrates the effectiveness of preprocessing and data augmentation in enhancing model robustness and reducing misclassification errors.The confusion matrix from the final model showed a substantial reduction in misclassification errors, particularly for red, maroon, and green vehicles, demonstrating the effectiveness of the targeted augmentation and dataset balancing strategies.

Metric	Before Classwise	After Classwise
Accuracy	73.5%	83.5%
Precision	71.1%	80.7%
Recall	69.6%	79.6%
F1 Score	70.3%	79.5%

TABLE III
PERFORMANCE COMPARISON BEFORE AND AFTER CLASSWISE AUGMENTATION USING YOLOv5

The confusion matrix from the final model showed a substantial reduction in misclassification errors, particularly for red, maroon, and green vehicles, demonstrating the

Fig. 8. Images With Detected Car Color

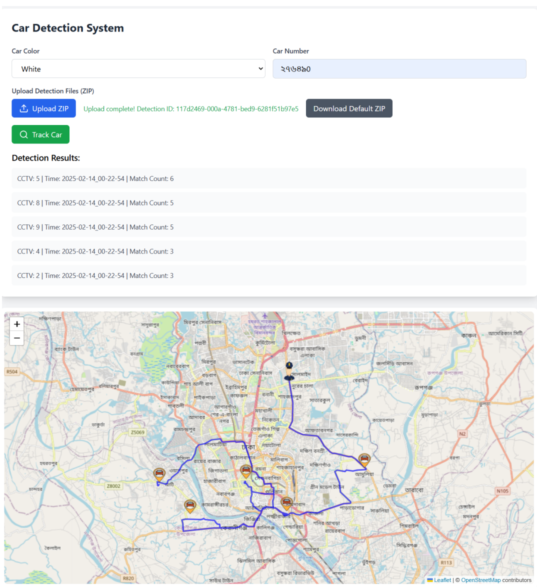

Fig. 9. Detected Car Path

effectiveness of the targeted augmentation and dataset balancing strategies.

Output Visualization and System Efficacy: Qualitative analysis of the system output confirms its capability in accurately detecting vehicles with class labels. Examples of model output visualization are shown in Figures 8 and 9.

V. Conclusion

Hit-and-run incidents remain a significant challenge in modern urban environments, often leaving victims without justice due to the inefficiencies of traditional investigative methods. Our proposed system aims to bridge the gap between manual analysis and automated surveillance by leveraging deep learning-based image processing techniques to enhance vehicle detection, tracking, and license plate recognition. By integrating YOLO-based models, our approach significantly improves the accuracy and efficiency of identifying suspect vehicles, even in cases involving low-quality CCTV footage or complex urban traffic conditions.

Our research highlights the importance of multi-camera tracking and dynamic suspect filtering, which enable law enforcement to reconstruct the movement of suspect vehicles across different surveillance points. Unlike existing methodologies that focus on static vehicle identification, our system dynamically updates suspect lists and generates real-time trajectory mappings to assist authorities in timely intervention. Through timestamp-based metadata integration and Google Maps API support, our system also provides a strategic tool for route optimization in suspect vehicle pursuit.

Despite the advantages of our approach, certain challenges remain, including the high cost of implementation, infrastructural limitations, and the need for extensive camera network coverage. Future research should explore optimizing computational efficiency, expanding the dataset for better generalization, and integrating additional AI-driven analytics to further improve accuracy and reliability. Additionally, policy-level support and inter-agency collaboration will be crucial in ensuring widespread adoption and effectiveness.

In conclusion,our AI-powered vehicle tracking system represents a significant step toward automating hit-and-run investigations, reducing manual workload, and enhancing the speed and accuracy of suspect identification. By addressing current limitations and continuously improving the system, this technology has the potential to revolutionize law enforcement methodologies, contributing to safer roads and more effective justice systems worldwide.

References

[1] S. Biswas, "Road traffic injuries: an emerging problem in bangladesh," *Faridpur Medical College Journal*, vol. 7, no. 1, pp. 5–5, 2012.

[2] Z. Zaman, M. H. Meraz, S. Siddike, F. N. Khan, M. U. Hasan, and M. M. Rahman, "Classifying cctv image for road accident analysis using deep learning techniques," in *2024 IEEE International Conference on Computing, Applications and Systems (COMPAS)*. IEEE, 2024, pp. 1–5.

[3] M. B. U. Khan, M. B. Aziz, M. O. Faruk, and M. I. A. Talukder, "Impact of cctv surveillance on crime prevention: A study in dhaka city," *Journal of Social Sciences and Humanities*, vol. 6, no. 2, pp. 48–59, 2020.

[4] G. Jacobs, A. Aeron-Thomas, and A. Astrop, "Estimating global road fatalities," 2000.

[5] N. Chen and Y. Chen, "Anomalous vehicle recognition in smart urban traffic monitoring as an edge service," *Future Internet*, vol. 14, no. 2, p. 54, 2022.

[6] W. Balid, H. Tafish, and H. H. Refai, "Intelligent vehicle counting and classification sensor for real-time traffic surveillance," *IEEE*

Transactions on Intelligent Transportation Systems, vol. 19, no. 6, pp. 1784–1794, 2018.
[7] A. Rehman, A. Malik, A. U. Rahman, S. Iqbal, and G. Tahir, "Cveh: A dynamic framework to profile vehicle movements to mitigate hit and run cases using crowdsourcing," 06 2021.
[8] X. Zhao, D. Dawson, W. A. Sarasua, and S. T. Birchfield, "Automated traffic surveillance system with aerial camera arrays imagery: Macroscopic data collection with vehicle tracking," *Journal of Computing in Civil Engineering*, vol. 31, no. 3, p. 04016072, 2017. [Online]. Available: https://ascelibrary.org/doi/abs/10.1061/%28ASCE%29CP.1943-5487.0000646
[9] B. Coifman, D. Beymer, P. McLauchlan, and J. Malik, "A real-time computer vision system for vehicle tracking and traffic surveillance," *Transportation Research Part C: Emerging Technologies*, vol. 6, no. 4, pp. 271–288, 1998. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0968090X98000199
[10] B. Tian, Y. Li, B. Li, and D. Wen, "Rear-view vehicle detection and tracking by combining multiple parts for complex urban surveillance," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 2, pp. 597–606, 2014.
[11] C.-Y. Hsu, L.-W. Kang, and H.-Y. Mark Liao, "Cross-camera vehicle tracking via affine invariant object matching for video forensics applications," in *2013 IEEE International Conference on Multimedia and Expo (ICME)*, 2013, pp. 1–6.
[12] M. Patil, S. Patil, A. Pawar, M. M. Bewoor, A. Kadam, U. Patkar, K. Wadare, and S. Sharma, "Vehicle number plate detection using yolov8 and easyocr," 07 2023, pp. 1–4.
[13] D. Vedhaviyassh, R. Sudhan, G. Saranya, M. Safa, and D. Arun, "Comparative analysis of easyocr and tesseractocr for automatic license plate recognition using deep learning algorithm," in *2022 6th International Conference on Electronics, Communication and Aerospace Technology*, 2022, pp. 966–971.