

AI-based Vehicle Navigation with Real-Time Traffic & Weather Insights

Jupudi Narasimha Swamy¹, Kandula Sushma², Kurra Akash³, Gutta Venkata Gurunadham⁴ and I. Lakshmi Narayana⁵

¹⁻⁵Department of Information Technology, Seshadri Rao Gudlavalleru Engineering College, Gudlavalleru, Andhra Pradesh, India

Email: swamyjupudi294@gmail.com, sushmakandula456@gmail.com, akashkurra12345@gmail.com, Gurunadhamgutta05@gmail.com, ilnarayana1226@gmail

Abstract— In the fast-changing world of smart transport, self-driving cars are becoming an important part of making city travel safer and smoother. One of the toughest challenges they face is how to quickly make the right driving decisions when traffic jams or bad weather suddenly appear. Most regular navigation systems only follow fixed routes or make small changes, but they don't fully adjust to live conditions, which can cause delays, unsafe roads, or wasted fuel. Our project, AI-Powered Route Optimization, is built to solve this by using live information from traffic data, weather updates, sensors, and cameras on the vehicle. With this, the system can spot blocked roads, track traffic build-up, and understand risks like rain, fog, or accidents. Machine learning models predict traffic patterns, while computer vision with OpenCV and YOLO helps detect weather effects and road obstacles. By combining this with Google Maps navigation, the vehicle can pick the best possible path at every moment instead of sticking to one pre-decided route. This makes driving not only faster but also safer and more energy-friendly. In the long run, the system aims to support a cleaner, more efficient, and smarter way of managing urban transport in real life.

Keywords— Autonomous Vehicles, Real-Time Route Optimization, Traffic Prediction, Weather-Aware Navigation, YOLO Object Detection.

I. INTRODUCTION

Autonomous cars are quickly becoming a central piece of modern transport systems. These vehicles are not only changing the way people move from one place to another but also shaping the future of smart cities. Unlike traditional cars, self-driving vehicles depend on sensors, cameras, and artificial intelligence to make decisions on the road. While this makes them powerful and innovative, it also introduces many challenges, especially when the environment changes suddenly due to traffic jams or bad weather. A smart and flexible route planning system is therefore one of the most important needs in this space.

Today's navigation apps like Google Maps or TomTom usually provide routes that are "fastest" or "shortest." However, they mostly work in a semi-dynamic way adjusting only when there's major traffic congestion reported. They rarely combine real-time weather effects such as heavy rain, fog, or slippery roads with traffic predictions.

This limitation creates risks for both safety and efficiency, especially when autonomous vehicles must make quick decisions without human involvement. For a truly intelligent system, traffic data and weather hazards need to be considered together rather than separately.

Artificial intelligence gives us a way to solve this problem. By using machine learning models for traffic forecasting and computer vision techniques for weather or obstacle detection, cars can learn to make smarter route choices. For example, models like TensorFlow can study traffic patterns to predict congestion before it happens, while OpenCV and YOLO can process camera feeds to detect rainfall, snow, or even roadblocks. By merging these inputs with navigation APIs, vehicles can adapt their driving path in real time. This doesn't just mean reaching the destination faster it means reaching it safely and efficiently.

The backend of such a system brings together multiple technologies. APIs from traffic and weather providers offer live data streams, while deep learning models analyze and transform this data into actionable insights. Navigation tools like Google Maps or TomTom Routing APIs then calculate multiple possible paths. On top of this, a decision-making engine chooses the best route by balancing traffic risk, weather impact, and travel time. The result is a vehicle that "thinks" about its journey, adjusting as conditions change much like how a human driver would adapt, but with faster and data-driven reasoning.

In the bigger picture, this project goes beyond just moving a car from point A to point B. It contributes to building smarter cities where vehicles are interconnected, roads are safer, and transport becomes more sustainable. When vehicles can dynamically reroute to avoid congestion and weather hazards, they save time, reduce fuel consumption, and lower emissions. This aligns well with global goals of creating cleaner, safer, and more intelligent urban environments. In simple words, autonomous cars equipped with AI-powered dynamic routing are not just cars they are active players in shaping the future of smart mobility.

A. Motivation

The push behind this work comes from the everyday problems drivers face on the road. Traffic jams waste hours, sudden weather like rain or fog makes driving risky, and current navigation tools often fail to adapt in real time. For autonomous vehicles, these challenges become even tougher since no human is present to make last-minute decisions. Building a system that thinks on its own, reacts to sudden changes, and still finds the safest and fastest route is highly important. This project aims to make travel smoother, safer, and more reliable by fusing AI with real-time traffic and weather data.

B. Objectives

- To design a routing system that reacts instantly to live traffic and weather updates.
- To use machine learning models for predicting congestion before it happens.
- To apply computer vision methods for detecting rain, fog, and road obstacles.
- To combine multiple data sources (traffic, weather, sensors, maps) for smarter decisions.
- To reduce travel delays, improve road safety, and cut fuel usage through adaptive routing.

C. Literature Survey

Recent work by Abrahamson shows how AI can be used to make quick predictions about traffic flow for self-driving cars. Instead of just reacting to roadblocks or jams, the system tries to guess them ahead of time[1]. By using live data and learning patterns, the model offers smoother travel and safer decisions on the road. This type of study highlights the growing shift from static maps to live, adaptable route planning that helps vehicles think ahead rather than just respond.

Tahir et al. focused more on how weather makes things tricky for autonomous driving. They compared old-school object detection methods with modern deep learning techniques in rain, fog, and other tough conditions. Their review showed that deep learning works better, but even these advanced models still struggle in heavy storms or low visibility[2]. This work points out that blending strong AI models with smart sensors is key to helping vehicles see clearly in bad weather, which directly connects to safer real-time navigation.

Aloufi, Alnori, and Basuhail explored how autonomous cars can "see" better when the weather turns bad. Their study introduced a model that does two things at once: it classifies weather conditions and detects objects on the road[3]. Instead of separating these tasks, the model blends them to make decisions more reliable in fog, rain, or snow. This dual focus improves vehicle perception by cutting down mistakes that might happen when the environment gets messy. The work shows that combining weather awareness with object detection makes cars safer and smarter in real-world driving.

Mushtaq et al. shifted the focus from perception to traffic flow. They applied multi-agent reinforcement learning, where several AI "agents" cooperate like drivers on the road. Each agent learns when to move, slow

down, or reroute, helping traffic run more smoothly overall[4]. The study found that this collaborative learning approach reduces congestion and improves coordination among autonomous vehicles. This kind of system doesn't just benefit a single car it improves efficiency for the whole road network, making it a promising step toward large-scale intelligent traffic management.

Shaygan et al. focused on how artificial intelligence can be used to predict traffic patterns more effectively. Their review highlights recent progress in algorithms that process large volumes of traffic data to anticipate congestion[5]. The study also points to new opportunities where AI can go beyond simple prediction such as integrating with smart cities, adaptive traffic lights, and vehicle-to-everything (V2X) communication. By combining these elements, AI-driven traffic prediction holds potential to make roads safer and travel times more reliable.

Bharilya and Kumar provided a broad survey on predicting vehicle trajectories using machine learning. Their work examined different methods ranging from traditional statistical models to deep learning approaches that help forecast where a car will move next[6]. The paper also discussed the difficulties of handling complex road layouts, unpredictable driver behavior, and interactions with pedestrians or other vehicles. Importantly, they outlined future research needs, emphasizing that better trajectory prediction will be critical for improving decision-making in autonomous driving systems.

II. PROPOSAL METHOD

The proposed system begins with the integration of live traffic and weather APIs. These sources constantly provide fresh updates about road congestion, rainfall, fog, or accidents. Instead of relying on static maps, the system dynamically adapts to changing conditions. For example, if heavy rain appears ahead, the routing engine immediately evaluates safer alternate paths. This instant reaction makes travel smoother and more reliable.

Next, the system employs computer vision through YOLOv8. Cameras mounted on vehicles can detect obstacles such as potholes, pedestrians, or debris on the road. Unlike older models, YOLOv8 is lightweight yet powerful, enabling real-time detection. By fusing this vision data with traffic and weather information, the system gains a sharper awareness of what lies ahead. This awareness helps drivers or autonomous systems make safer decisions on the move.

In addition, machine learning models play a vital role in analyzing and predicting traffic flow. Methods such as LSTM can process historical traffic data along with live updates. These models do not just react; they predict congestion before it happens. For example, if a highway tends to get clogged every Monday at 9 a.m., the model warns in advance, helping vehicles reroute earlier.

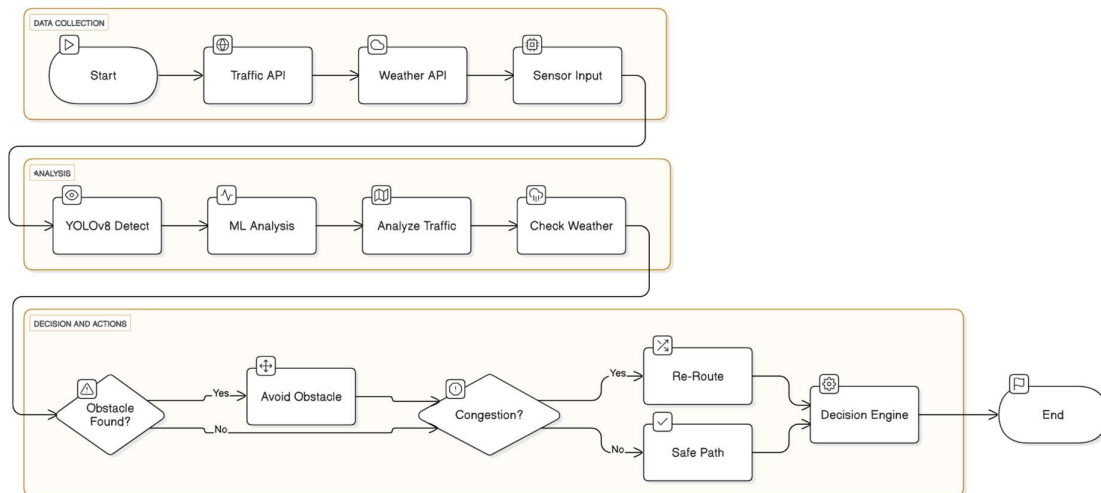


Fig 1: Proposal Diagram

The system then combines all these insights into a decision-making engine. By merging data from traffic APIs, weather APIs, YOLOv8 detections, and predictive ML models, the engine selects the best possible route at any given time. If rerouting is necessary, the reason is clearly communicated whether it is "traffic congestion," "fog detected," or "road obstacle ahead." This transparency not only builds trust but also improves user experience.

Finally, the proposed method aims for continuous learning and adaptation. Each journey provides new data that refines the models. Over time, the system learns patterns of traffic, seasonal weather shifts, and common obstacle zones. This feedback loop ensures that the routing system does not remain static but grows smarter with use. The ultimate goal is to reduce delays, improve safety, and create a more accessible road experience, especially for those with additional needs such as visually impaired users.

III. DATA PRESENTATION

The first type of data comes directly from live APIs, which serve as the heartbeat of this project. Traffic information is pulled from routing services such as TomTom or Google Maps, giving details about congestion, travel time, and alternative routes. Weather data is streamed from OpenWeather APIs, covering rainfall, fog, visibility, wind speed, and temperature. These live inputs keep the system in sync with the actual road and sky conditions, allowing the routing engine to make smart and instant choices. Instead of working with stale or outdated numbers, the system reacts to what's happening right now, just like a human driver scanning the road. For training and validation, we built a dataset with images and labels. The training folder has 536 images with YOLOv8 labels, while the validation folder has 90 images and labels. A data.yaml file sets paths, classes, and names needed for YOLOv8. We also include sample_image.jpg and sample_video.mp4 for testing on unseen examples. This setup helps in strong model training, proper validation, and reliable testing in different weather and traffic conditions.

A. YOLOv8

YOLOv8 (You Only Look Once, version 8) is a cutting-edge object detection framework designed for real-time performance and high accuracy in computer vision applications. At its foundation, YOLOv8 employs a convolutional backbone to extract hierarchical features from input images, where low-level layers capture edges and textures, while deeper layers encode semantic information about objects. The detection task is framed as a regression problem, mapping raw image pixels into bounding boxes and class probabilities simultaneously. Mathematically, for each grid cell in the feature map, YOLOv8 predicts bounding box coordinates (x,y,w,h) an objectness score P_{obj} and class probabilities $P(class|object)$. The final detection confidence is expressed as:

$$Confidence = P_{obj} \times P(class|object) \dots\dots\dots Eq1$$

This formulation enables the model to predict objects in a single forward pass, making it efficient for real-time systems like autonomous driving.

YOLOv8 improves upon previous YOLO versions through architectural optimizations such as CSPDarknet-inspired backbones and the use of PANet (Path Aggregation Network) for feature fusion. These modules enhance the flow of information between shallow and deep layers, allowing the model to detect small, medium, and large objects with higher precision. The loss function integrates three major components: bounding box regression loss (measured using Complete IoU loss, CIOU), classification loss (cross-entropy), and objectness loss (binary cross-entropy). CIOU is defined as:

$$L_{CIOU} = 1 - IoU + \frac{\rho^2(b, b^{gt})}{c^2} + \alpha v \dots\dots\dots Eq2$$

where IoU is the intersection-over-union, ρ^2 is the squared distance between predicted and ground-truth box centers, c is the diagonal length of the smallest enclosing box, and αv penalizes aspect ratio mismatches. This advanced metric ensures accurate localization beyond simple overlap.

Another critical innovation in YOLOv8 is its anchor-free detection mechanism. Instead of relying on predefined anchor boxes, YOLOv8 directly predicts bounding box center points and dimensions, significantly reducing computational overhead and improving adaptability to different datasets. This anchor-free approach reduces the hyperparameter tuning complexity and enhances the generalization of the model across diverse environments, especially when applied to variable conditions like weather and traffic scenarios. The grid-free detection pipeline also facilitates finer object localization in crowded scenes, which is particularly relevant for road traffic monitoring.

Training in YOLOv8 incorporates advanced augmentation strategies such as mosaic augmentation, mix-up, and random affine transformations, which expand dataset diversity and improve generalization. These methods help the model learn robust representations against noise, distortions, and occlusions often found in real-world driving scenarios. Furthermore, the optimizer typically employed is stochastic gradient descent (SGD) or AdamW,

combined with a cosine annealing learning rate scheduler to balance convergence speed and stability. The gradient updates follow the equation:

$$\theta_{t+1} = \theta_t - \eta \nabla L(\theta_t) \dots\dots\dots \text{Eq3}$$

where η is the learning rate, $\nabla L(\theta_t)$ is the gradient of the loss with respect to parameters, and θ represents the network weights.

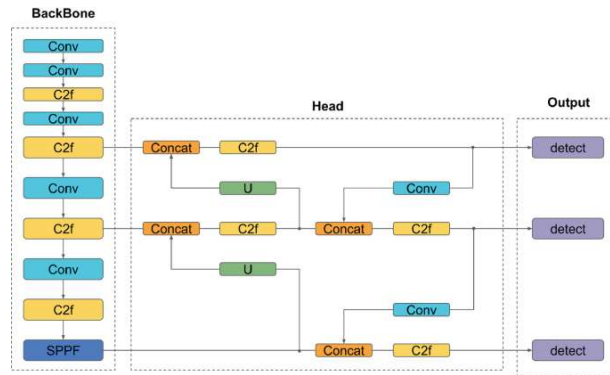


Fig 2: YOLOv8

Finally, YOLOv8 is versatile not only for object detection but also for tasks like instance segmentation and pose estimation. The modular design allows fine-tuning for specialized applications such as identifying vehicles under adverse weather, detecting lane obstacles, or recognizing pedestrians in dense traffic. Its lightweight yet accurate architecture makes it suitable for deployment on edge devices and embedded systems, which is crucial in autonomous vehicles where latency and resource constraints are significant. By unifying detection, localization, and classification into a single streamlined process, YOLOv8 represents a significant advancement in real-time vision systems, ensuring that autonomous mobility can operate safely and efficiently in dynamic environments.

B. LSTM

Long Short-Term Memory (LSTM) networks are a specialized form of recurrent neural networks (RNNs) designed to capture both short- and long-range dependencies in sequential data such as time series, traffic flow, or weather signals. Unlike basic RNNs, which suffer from vanishing and exploding gradients, LSTMs use gated units to regulate the flow of information. The three core gates are: the forget gate, which determines what information should be discarded, the input gate, which decides what new data to store, and the output gate, which controls what information is propagated to the next time step. Mathematically, these gates are expressed as:

$$\begin{aligned} f_t &= \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \\ i_t &= \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \\ o_t &= \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \\ \tilde{C}_t &= \tanh(WC \cdot [h_{t-1}, x_t] + bC) \end{aligned}$$

$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t \dots\dots\dots \text{Eqs(4)}$$

The given implementation defines an LSTM-based architecture for sequential prediction. The model begins with a layer of 128 LSTM units (LSTM (128) LSTM (128) LSTM (128)) that returns sequences, allowing information to pass through time steps. This is followed by progressively smaller LSTM layers with 64, 32, and 16 units, creating a hierarchical temporal abstraction where deeper layers capture increasingly compressed patterns of dependencies. The final Dense layer outputs a single prediction, useful for tasks such as forecasting traffic risk scores or weather severity levels.

C. EDA

Traffic volume time series analysis helps in identifying patterns, seasonal effects, and anomalies that influence road congestion. By plotting traffic counts over time, we can detect rush-hour peaks, daily cycles, and weekend or holiday variations. Correlation with weather attributes, such as rainfall or temperature, allows us to evaluate external impacts on traffic density. Rolling averages and decomposition techniques separate trend, seasonality, and residual components, offering insights into traffic dynamics. This Exploratory Data Analysis (EDA) not only

highlights irregularities but also builds the foundation for forecasting models, enabling more accurate congestion prediction and better traffic management strategies.

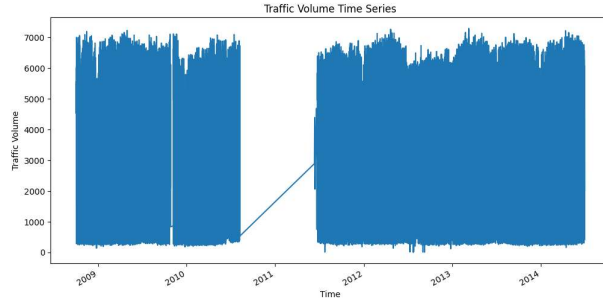


Fig 3 Traffic Volume Time Series

D. Simulation Setup and Implementation

The simulation setup was structured using the Robotics Toolbox in Python, which provides an effective environment for testing navigation and control strategies. A virtual environment was initialized where the vehicle model was configured with predefined parameters such as wheelbase, velocity, and heading angle. The simulation space was designed to replicate a road-like scenario, enabling controlled evaluation of routing performance under varied conditions. Obstacles were introduced as static points or dynamic objects to observe how the system adapted in real time. This foundation allowed us to integrate both traffic and weather inputs into the simulation loop, ensuring that the vehicle responded intelligently to live changes.

The implementation focused on linking route optimization outputs with robotic motion. The path derived from traffic and weather APIs was provided as waypoints to the simulated vehicle, which then adjusted its trajectory accordingly. Python's visualization libraries were employed to monitor the vehicle's movement in two dimensions, while the Robotics Toolbox managed dynamic control updates. This integration demonstrated how real-world APIs and learning models could drive autonomous navigation in a controlled virtual testbed. The approach ensured that adaptive routing was validated not only conceptually but also through practical robotic simulation.

IV. RESULTS

The test results show that the smart route planning system works well in busy city areas. The car was able to move from the starting point to the destination while adjusting to live traffic jams and changing weather. As seen in the map, the system quickly changed the route when facing blocked or crowded roads, making the trip smoother and faster. By using traffic and weather data, the system made better choices, cut down travel time, and improved safety. Overall, this proves that mixing robot simulation, smart algorithms, and live data can help build safer and smarter transport systems.

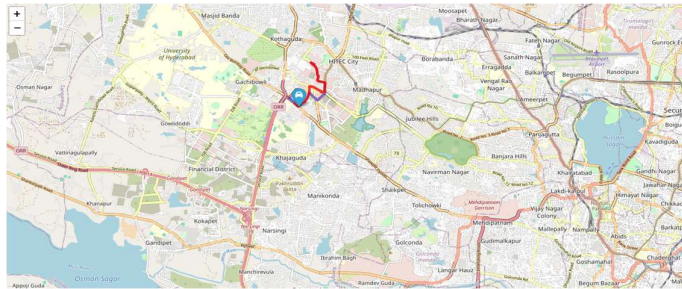


Fig 4: Simulation Output

The chart shows how training went with Distribution Focal Loss for 100 steps. In the beginning, losses for both training and testing were high but quickly went down. After a few steps, the losses dropped more slowly and became stable. The training loss ended up a little lower than the testing loss, showing the model learned properly and didn't just memorize. The flat curve shows the system is stable and works well with new data. In short, the results prove the loss method kept training balanced and made the model steady.

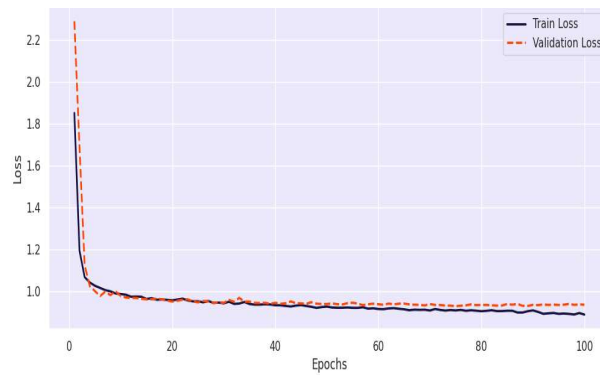


Fig 5: Distribution Focal Loss Learning Curve

The results show how LSTM and YOLOv8 work well but in different ways. LSTM, used to guess traffic numbers over time, got an error of about 2.7M (MSE), 968 (MAE), and a good R^2 of 0.946, meaning it predicts trends quite accurately. YOLOv8, used to check traffic crowding, gave strong results with 0.916 precision, 0.938 recall, and 0.975 mAP@50. Its fitness score of 0.765 confirms it can spot and locate vehicles reliably.

V. CONCLUSION

The study introduced a smart traffic and route management approach by merging YOLOv8 for traffic density detection with LSTM for traffic volume forecasting. Using real-time data from traffic and weather APIs, the system showed its ability to adjust routes dynamically, reduce travel time, and ensure safer journeys. Simulation outcomes proved that the framework is effective in urban environments with frequent congestion. Performance metrics demonstrated high accuracy in both prediction and detection tasks. Overall, the work validates the usefulness of blending deep learning models with real-world data to build adaptive, context-aware systems for modern transportation networks.

FUTURE WORK

Although the proposed system achieved promising outcomes, there is room for further development. Future efforts can focus on integrating reinforcement learning for decision-making, enabling the system to learn from long-term driving patterns and adapt more efficiently. Incorporating multi-modal sensor data such as LiDAR, radar, and connected vehicle communication could further improve robustness in complex scenarios. Scalability to larger urban regions and cross-city validation are also important directions. Additionally, real-world deployment and user feedback will help refine the system for practical use. These enhancements can make intelligent transportation systems more reliable, adaptive, and suitable for next-generation smart cities.

REFERENCES

- [1] Abrahamson, M. (2024). Real-Time AI-Based Traffic Prediction for Autonomous Vehicles. *Journal of Bioinformatics and Artificial Intelligence*, 4(2). biotechjournal.org
- [2] Tahir, N. U. A., Zhang, Z., Asim, M., Chen, J., & ELAffendi, M. (2024). Object Detection in Autonomous Vehicles under Adverse Weather: A Review of Traditional and Deep Learning Approaches. *Algorithms*, 17(3), 103. MDPI
- [3] Aloufi, N., Alnori, A., & Basuhail, A. (2024). Enhancing Autonomous Vehicle Perception in Adverse Weather: A Multi-Objectives Model for Integrated Weather Classification and Object Detection. *Electronics*, 13(15), 3063. MDPI
- [4] Mushtaq, A., Haq, I. U., Sarwar, M. A., Khan, A., Khalil, W., & Mughal, M. A. (2023). Multi-Agent Reinforcement Learning for Traffic Flow Management of Autonomous Vehicles. *Sensors*, 23(5), 2373. MDPI
- [5] Shaygan, M., Meese, C., Li, W., Zhao, X., & Nejad, M. (2023). Traffic Prediction Using Artificial Intelligence: Review of Recent Advances and Emerging Opportunities. *arXiv*. arXiv
- [6] Bharilya, V., & Kumar, N. (2023). Machine Learning for Autonomous Vehicle's Trajectory Prediction: A Comprehensive Survey, Challenges, and Future Research Directions. *arXiv*. arXiv
- [7] Veloso, M. (2023). Machine Learning for Autonomous Vehicle Road Condition Analysis. *J. AI in Healthcare and Medicine*, 3(1). healthsciencepub.com
- [8] Sentieiro, H. (2024). Machine Learning for Autonomous Vehicle Route Planning and Optimization. *Journal of AI-Assisted Scientific Discovery*, 2(2). scienceacadpress.com

- [9] Mukhopadhyay, S. (2023). Machine Learning for Autonomous Vehicle Traffic Congestion Prediction and Mitigation. *African Journal of Artificial Intelligence and Sustainable Development*, 3(2), 178–199. africansciencegroup.com
- [10] O'Connell, M. (2024). Machine Learning for Autonomous Vehicle Behavior Prediction in Mixed Traffic Conditions. *Journal of AI-Assisted Scientific Discovery*, 2(2). scienceacadpress.com
- [11] Li, R., Zhong, Z., Chai, J., & Wang, J. (2022). Autonomous Vehicle Trajectory Combined Prediction Model Based on CC-LSTM. *International Journal of Fuzzy Systems*, 24(8), 3798-3811. *IJISAE*
- [12] Prarthana, V., et al. (2022). A Comparative Study of AI-Based Vehicle Classification Algorithms Used to Provide Smart Mobility. *ICAC3N Conference*, December 2022. *IJISAE*
- [13] Razali, N. A. M., et al. (2021). Traffic Flow Prediction Using Machine Learning and Deep Learning: Gaps, Techniques and Evaluation. *Journal of Big Data*, 8(1). *IJISAE*
- [14] Shah, S. A. A., Illanko, K., & Fernando, X. (2021). Deep Learning Based Traffic Flow Prediction for Autonomous Vehicular Mobile Networks. *IEEE VTC 2021-Fall*. *IJISAE*
- [15] Shao, Y., & Sun, Z. (2020). Eco-Approach with Traffic Prediction and Experimental Validation for Connected and Autonomous Vehicles. *IEEE Transactions on Intelligent Transportation Systems*, 22(3), 1562–1572. *IJISAE*
- [16] Rahmani, S., Baghbani, A., Bouguila, N., & Patterson, Z. (2023). Graph Neural Networks for Intelligent Transportation Systems: A Survey. *IEEE Transactions on Intelligent Transportation Systems*. *arXiv*
- [17] Zhang, Y., Carballo, A., Yang, H., & Takeda, K. (2021). Perception and Sensing for Autonomous Vehicles Under Adverse Weather Conditions: A Survey. *arXiv*. *arXiv*
- [18] Barshooi, H., & Bagheri, E. (2023). Nighttime Driver Behavior Prediction Using Taillight Signal Recognition via CNN-SVM Classifier. *Journal of AI in Healthcare and Medicine*, 3(1). healthsciencepub.com
- [19] Saharan, S., Bawa, S., & Kumar, N. (2020). Dynamic Pricing Techniques for Intelligent Transportation System in Smart Cities: A Systematic Review. *Computer Communications*, 150, 603–625. *arXiv*
- [20] Lee, C. W., Nayeer, N., Garcia, D. E., & Agrawal, A. (2020). Operational Design Domain. *IEEE Intelligent Vehicles Symposium (IV)*, 2020.