ISSN: 2583-9888
DOI: doi.org/10.51219/JAIMLD/avani-krunal/225

URF PUBLISHERS
connect with research world

Journal of Artificial Intelligence, Machine Learning and Data Science

https://urfpublishers.com/journal/artificial-intelligence

Vol: 2 & Iss: 1

Research Article

Dashcam-Eye: Federated Learning Based Smart Dashcam Based System for Automotives

Avani Dave* **and Krunal Dave**

Citation: Dave A, Dave K. Dashcam-Eye: Federated Learning Based Smart Dashcam Based System for Automotives. *J Artif Intell Mach Learn & Data Sci 2024*, 2(1), 942-945. DOI: doi.org/10.51219/JAIMLD/avani-krunal/225

Received: 02 March, 2024; **Accepted:** 18 March, 2024; **Published:** 20 March, 2024

***Corresponding author:** Avani Dave, USA, E-mail: daveavani@gmail.com

ABSTRACT

With technological advancements automotive industry is witnessing new era of transition to self-driving vehicle for everything. Self-driving cars for personal and commercial use are right across the corner to become full reality. Along with self-driving drone and robots for delivery of food and merchandise have become reality. With these advancements the safety and security of the self-driving vehicles have emerged as major concern for current times. To this end, this work proposes DashCamEye Federated learning based Smart Dash cam which continuously records and then processes video imagery while in route. The simulation results of DashCamEye based system with multiple scenarios shows promising results and opens a new research direction to explore to provide safe and informed route prediction for self-driving vehicles of next generation.

Keywords: Automotive chiplet architecture, Next-generation vehicular systems, Autonomous driving, Fusion sensors, ADAS, Infotainment, Gem5, Chiplet, Mcpat

1. Introduction

Self-driving vehicles have emerged as a revolutionary technology, and it has transformed the modern-day transportation landscape. Modern-day automotives have a large network of computing, sensing, and processing engines. These systems are connected to the internal and external network for exchanging boot time and runtime critical information which are used for making informed decisions at different stages. These devices have built in Global Positioning Systems (GPS), satellite networks or a connected smart application (central controller / tablet screen / phones) to help users navigate the direction from point A to point B. There has been significant research and development to aid in autonomous driving and route planning in recent years[1-6].

Self-driving cars can potentially reduce traffic accidents caused by human errors[7] and provide more safer roads to users including pedestrians[8]. The routes used by self-driving vehicles will be based on shortest distance, toll, freeway, time of the day, traffic etc. Some of the latest applications uses machine learning and AI algorithms to get runtime updates such as accidents, road work and traffic slow alerts. However, routing applications lacks in accounting for safety, user preference and current surrounding metrics such as safety (break-ins[9], vandalism[10], porch-pirates[11] etc.), pollution, ease of parking, emergency situations[12], food choices, etc. Given a city and surrounding areas have "better" and "worse" regions and local humans will be aware of the neighborhood and dynamic conditions but the self-driving vehicles (cars, robots, delivery drones etc.) currently lack in such dynamic route optimization. To this end, this work presents a novel federated learning combined with deep reinforcement learning based approach to help better route planning with dynamic changes in metrics. This work also presents promising simulation results with optimized algorithm.

(Figure 1) depicts various non conventional sources of data collection my various sensors in self-driving cars, drones or robots. These dynamic sources of information provides realtime conditions for optimized route predictions. DashCamEye uses this data as weights on the routing map and applies coordinated policy optimization for better decision making with reinforcement learning. DashCamEye algorithm collects the event markers and send it to the could server with timestamp and geolocation information. These cloud sourced information will be used by DashCamEye server to perform federated learning and optimize the model over time. Thus, DashCamEye simulation combines federated learning approach with coordinated policy optimization. DashCamEye will not only provide better dynamic decision making at runtime but also it will aid in better route optimization to avoid certain situations such as crowd gathering, road blockage, criminal activities, porch pirates etc for the self-driving vehicles.

We propose to build a smart dashcam device which continuously records and then processes video imagery while in route. The results of the fast processing and labelling, once runs through a more intensive image processing verification step, are uploaded to a central web server which holds a dynamically updateable database of road segment (edge) information and corresponding calculated weights and labels for the area graph. The graph is continuously updated based on the information received from different sources (dashcams and publicly available information) and can be used to display a route map with current predicted conditions for the fleet vehicle operators and optimized, more efficient and safer routing, using the estimated weights and markings for each edge of the area graph.

2. Related Work

With the advancement in self-driving automotives industry many researchers have explore different Machine Learning (ML) and Artificial Intelligence (AI) techniques for enhancing self-driving experience. Multi Agent Reinforcement Learning (MARLA) has emerged as powerful approach to solve complex decision-making problems. The typical task settings are divided in three categories namely fully cooperative task which focuses on communication[13,14] and credit assignment[15-17]. Second category is competitive tasks which focuses on meaningful opponents[18,19]. Third is mixed approach[20-22]. CoPo[23] presents cooperative policy optimization simulator for dynamic tasks handling at runtime.

Second key aspect of the related work is to find suitable traffic simulator for evaluating the proposed technique. Various traffic flow simulators such as CARLA[24], SUMO[23], CityFlow[25] and FLOW[26] uses RL agents to steer the low-level controllers for investigating specified traffic conditions. SMARTS[27] evaluates the interaction between social vehicles dynamic traffic environments and RL agents. Maps based application such as Google maps, apple map, waze. Safety and neighbourhood watch apps such as citizen, crime-alert, neighbourhood watch etc, are few examples of route mapping apps that uses runtime reinforcement learning and adds weights and biases on the route selection. However, they do not count of cooperation-based policy optimization which from the simulation results aids in the safe self-driving vehicle experience.

3. Dash Cam Architecture

(Figure 2) Shows high-level system design of DashCamEye. The server will send initial model for training to the client application running on the self-driving vehicle upon installation. The automotive dashcam data such as and HD cameras readings are feed into federated learning based modified CoPo algorithm during runtime.

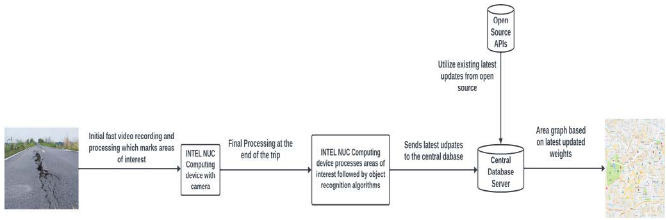

Figure 2: High-level system design for DashCamEye.

It will result in adding the weights and bias to the x and y coordinates of the geo location during runtime. The algorithm will take this weights and biasing with geo coordinates into account for cooperative policy optimization and agent will act based on not only past but current situation of the environment in which it is. The server collects the feedback of each client and use it for optimizing the training model and refeeds it to client application periodically. Thus, live updates of the geolocation and surrounding conditions are given to the modified CoPo algorithm. The policies are optimized on the client agent by combining local policy updates and the feedback received from the server as global policy updates.

The localized policy optimization is performed using Eq1

$$r_{i,t}^{N} = \frac{\sum_{j \in \mathcal{N}_{d_n}(i,t)} r_{j,t}}{|\mathcal{N}_{d_n}(i,t)|}, \text{ wherein } \mathcal{N}_{d_n}(i,t) = \{j : ||\text{Pos}(i) - \text{Pos}(j)|| \le d_n\}. \quad \ldots\ldots 1$$

Where, Ndn (i, t) defines the neighbourhood of agent i for the given radius dn at step t. The key reason for adding the global policy is to make route prediction accuracy high. As the vehicles are running in the road with dynamic surrounding and conditions of the neighbouring vehicles will also affect the reward given to the agent. When a consensus based global policy weights and biases were added to the routing algorithm. It improved the performance significantly. The co-ordinated and weighted rewards are defined using following equation:

$$r_{i,t}^{C} = \cos(\phi) r_{i,t} + \sin(\phi) r_{i,t}^{N}. \quad \ldots..2$$

Implementation of fast scenery recognition methods, whereby the dashcam continuously records video and marks select frames of the recorded video for further analysis, based on rapid image segmentation and object detection technology. For this, we can make use of existing image processing libraries such as PixelLib and e.g. public data sets such as Pascal VOC and YOLO algorithm approaches[2,3]. This will allow us to rapidly segment out the cars on the road in image frames obtained from the camera and estimate traffic density. We will also develop machine learning based techniques to quickly identify potential road issues (lane closures, large potholes, road-work equipment and personnel, etc.) for marking road segments with preliminary markings.

Implementation of more intensive machine learning based image analysis for processing and labeling marked frames from a predefined road and traffic condition characterization list (for example, route closure, lane restrictions, road work markings, traffic jam, etc.). In this step, we will take the segments with preliminary markings and run two-stage machine learning based approaches with higher computational load to ascertain the kind of condition present on the segment. For this, we can use object detection based on Convolutional Neural Networks, APIs from Google's TensorFlow and related developments[4,5]. This processing will be activated based on the readings of the motion detection module, when the system has ascertained that the vehicle has been stationary above a set threshold of time. It can also be manually disabled by the operator, if necessary.

Creation of an online database to store information for a defined area as referenced by location (lat, lon) and the time of day and data aggregation and prediction approaches to automatically predict information for all edges based on merged available data coming from different dashcam systems and publicly listed information. This information would be integrated together and used to predict a weight and any extra labeling information for each edge of the graph representing the area map for each (lat, lon, time) triplet associated with each edge (road segment) location. Development of a customized map display and routing methods based on a weighted area graph information from the database, via a web-based service development.

4. DashcamEye approach for adding weights and Biases

Implementation of routing methods based on a weighted graph approach, whereby each path between two defined locations consists of a set of weighted edges (corresponding to road segments). The weights of the edges are determined by factors such as distance, speed, road aspects, pollution, weather, and safety metrics. A UI interface with sliders for different aspects can be used to control the settings for the routing method to suggest routes based on user preferences (e.g., speed and safety, avoiding construction zones, etc.)

Creation of an online database to dynamically update the edge weights in a defined area as referenced by location (lat, lon) and the time of day and routing algorithms for two end points based on the weights from previous step. This database can utilize data from public sources and optionally from sensors placed on fleet vehicles, which regularly drive through the defined area.

Creation of a hardware module for the sensors to be mounted on fleet vehicles, including speed and camera sensors, and corresponding analysis algorithms. The cameras can analyze scenery and record e.g., instances of lane closures and construction presence and mark these aspects by uploading the information to the online database service.

5. Evaluation

The simulation was performed by modifying the CoPo simulator and including the weights and biases from the sensors RL learning. The modified algorithm and simulation were tested on following four scenarios for success rate and accuracy.

Table 1: Depicts the success rate of different techniques compared to DashCamEye.

Technique	Potholes	Intersection	Lane closures	accidents
IPO	64.67	60.47	72.43	83.5
MFPO	66	69.43	67.43	81
CoPo	72.6	78.34	74.21	75.6
DashCamEye	75.3	83.23	80.54	79.32
DASHCAM-EYE	77.2	85.43	81.24	79.89

Scenario 1: when Potholes and not favourable weather conditions were feed to the simulation motel with modified CoPo. The generated weight and biases were used for making policy-based prediction and optimized route selection the accuracy of the DashCamEye was increased by 3% compared to that of CoPo.

Scenario 2: Depicts when self-driving vehicle is on the road in the intersection with ongoing traffic with signal lights offs and the request for new route prediction comes to the client. The accuracy of this route and policy-based prediction with DashCamEye reaches 83.23 % correct results.

Scenario 3 and **Scenario 4** depicts road conditions blocked by accident and or construction on the road and lane closure as a result. In both the cases from the simulation results DashCamEye outperforms the other state-of-the-art techniques and provides better route and policy-based selection coordinated route prediction.

(Figure 3) depicts the map plugin based on shortest distance between two endpoints. **(Figure 4)** represents the Map with additional layer of CoPo based optimization for the route. **(Figure 5)** shows the DashCamEye provided route options based on safety weights and biases. The red colour road indicates the scratchy area and recommends the self-driving vehicle to potentially avoid it by choosing alternate green path.

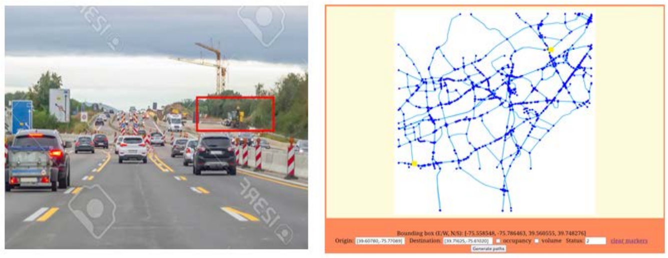

Figure 3: Sample object identification and web-based mapping interface.

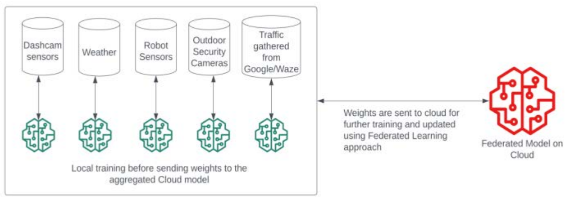

Figure 4: Dash Cam Eye route map with safety weight/bias.

6. Conclusions

With the increased commercialization of self-f=driving cars, robots, and drone the secure and safe route selection and dynamic decision making based on the current surrounding conditions becomes vital. Previous works have demonstrated use of reinforcement learning to tackle some of these issues. However, they lack in dynamic cooperated policy optimization to handle real world scenarios such as pothole, crime activity, roadblocks, constructions, weather impacts etc. DashCam Eye provides the dynamic cooperated policy-based decision optimization along with usage of federated learning enabled it to proliferate the learning to the could server network and with applied weights and biases this federated learning-based approach optimizes the training model to handle such situations in the future better. The simulation results shows that DashCamEye based route maps is successfully able to avoid the scratchy neighborhood during the specific time of the day and alerted the user to chose alternate route.

7. References

1. Yurtsever E, Lambert J, Carballo A, Takeda K. A survey of autonomous driving: common practices and emerging technologies. arXiv 2019.

2. Talpaert V, Sobh I, Kiran BR, Mannion P, Yogamani S, ElSallab A, Perez P. Exploring applications of deep reinforcement learning for real-world autonomous driving systems. arXiv 2019.
3. Kiran BR, Sobh I, Talpaert V, Mannion P, Sallab AAA, Yogamani S, Perez P. Deep reinforcement learning for autonomous ´ driving: A survey. arXiv 2020.
4. Schwarting W, Alonso-Mora J, Rus D. Planning and decisionmaking for autonomous vehicles. Annual Review of Control, Robotics, Autonomous Systems 2018.
5. Gonzalez D, Perez J, Milanes V, Nashashibi F. A review of motion planning techniques for automated vehicles. IEEE Transactions on Intelligent Transportation Systems 2015;17: 1135-1145.
6. Paden B, Cap M, Yong SZ, Yershov D, Frazzoli E. A survey of motion planning and control techniques for self-driving urban vehicles. IEEE Transactions on intelligent vehicles 2016;1: 33-55.
7. Ye F, Zhang S, Wang P, Chan C-Y. A survey of deep reinforcement learning algorithms for motion planning and control of autonomous vehicles. 2021 IEEE Intelligent Vehicles Symposium 2021; 1073-1080.
8. Shubbak, M. Self-Driving Cars: Legal, social, and ethical aspects. SSRN 2013.
9. Rasouli A, Tsotsos JK. Joint Attention in Driver-Pedestrian Interaction: from Theory to Practice. Cornell University 2018.
10. SFO crowd sets self-driving car on fire.
11. Ramev J. Robbing delivery robots is now a thing. Auto week 2023.
12. https://www.ktalnews.com/news/consumer-reports/how-ai-and-drones-will-help-prevent-porch-pirates-after-packages-arrive/
13. https://ktla.com/news/california/company-refutes-claim-that-self-driving-cars-blocked-california-ambulance-led-to-victims-death/#:~:text=According%20to%20the%20San%20Francisco,crews%20from%20reaching%20the%20hospital.
14. Foerster JN, Assael YM, De Freitas N, Whiteson S. Learning to communicate with deep multi-agent reinforcement learning. arXiv 2016.
15. Vanneste S, Vanneste A, Mets K, et al. Learning to communicate using counterfactual reasoning. arXiv 2020.
16. Yali Du, Lei Han, Meng Fang, Ji Liu, Tianhong Dai, and Dacheng Tao. LIIR: Learning individual intrinsic reward in multi-agent reinforcement learning. Advances in Neural Information Processing Systems 2019;32: 4403-4414.
17. Foerster J, Farquhar G, Afouras T, Nardelli N, Whiteson S. Counterfactual multi-agent policy gradients. Proceedings of the AAAI Conference on Artificial Intelligence 2018;32.
18. Rashid T, Samvelyan M, Schroeder C, Farquhar G, Foerster J, Whiteson S. Qmix: Monotonic value function factorisation for deep multi-agent reinforcement learning. International Conference on Machine Learning 2018; 4295-4304.
19. Vinyals O, Babuschkin I, Czarnecki WM, et al. Grandmaster level in starcraft ii using multi-agent reinforcement learning. Nature 2019;575: 350-354.
20. Heinrich J, Silver D. Deep reinforcement learning from self-play in imperfect-information games. arXiv 2016.
21. Leibo JZ, Zambaldi V, Lanctot M, Marecki J, Graepel T. Multi-agent reinforcement learning in sequential social dilemmas. arXiv 2017.
22. Lowe R, Wu Y, Tamar A, Harb J, Abbeel P, Mordatch I. Multi-agent actor-critic for mixed cooperative-competitive environments. Advances in neural information processing systems 2017.
23. Schwarting W, Pierson A, Alonso-Mora J, Karaman S, Rus D. Social behavior for autonomous vehicles. Proceedings of the National Academy of Sciences 2019;116: 24972-24978.
24. Behrisch Michael, et al. SUMO-simulation of urban mobility: An overview. SIMUL 2011, the third international conference on advances in system simulation. ThinkMind 2011.
25. Alexey D, Ros G, Codevilla F, Lopez A, Koltun V. CARLA: An open urban driving simulator. Conference on robot learning 2017.
26. Zheng T, Naphade M, Liu M-Y, et al. Cityflow: A city-scale benchmark for multi-target multi-camera vehicle tracking and re-identification. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition 2019.
27. Wu C, Kreidieh A, Parvate K, Vinitsky E, Bayen AM. Flow: Architecture and benchmarking for reinforcement learning in traffic control. arXiv 2017; 10.
28. Zhou M, Luo J, Villella J, et al. Smarts: Scalable multiagent reinforcement learning training school for autonomous driving, arXiv 2020.