OP-CAS: Collision Avoidance with Overtaking Maneuvers

Eun Sang Cha
Computer Science
Univ of California, Los Angeles
California, USA
gotenkscha@gmail.com

Kee-Eung Kim
Computer Science
Korean Advanced Institute of Science and Technology
Daejeon, South Korea
kekim@cs.kaist.ac.kr

Stefano Longo
Automotive Mechatronics
Cranfield University
Cranfield, United Kingdom
s.longo@cranfield.ac.uk

Ankur Mehta
Electrical Engineering
Univ of California, Los Angeles
California, USA
mehtank@ucla.edu

***Abstract*—This paper presents a novel collision avoidance system for autonomous vehicles based on overtaking procedures. The proposed Overtaking Procedure for Collision Avoidance Systems (OP-CAS) takes a behavioral cloning-based approach which uses images obtained out of a low cost monocular camera. The algorithm selectively records the expert's corrective driving behavior during data collection. This is performed recording oscillatory driving behavior when the vehicle is returning to the center of the lane. This data augmentation method addresses the issue of covariate shift commonly found in behavioral cloning methods. This approach is computationally inexpensive, making it a viable option for real time embedded deployment. A feasibility study was performed with two remotely controlled scaled vehicles as a proof of concept. Results showed that when two expert drivers demonstrated overtaking behaviors for data collection, even a small dataset was sufficient to model the overtaking sequence. The overtaking maneuvers were deployed in real time on 1/8th scale RC platforms, validating OP-CAS for civilian vehicle safety applications.**

***Index Terms*—Advanced Vehicle Safety Systems, Autonomous Vehicles, Convolutional Neural Networks**

I. Introduction

In efforts to mitigate the recent vehicle accidents occurring with autonomous vehicle testing, collision avoidance systems (CAS) are becoming an active topic of research. Current autonomous vehicle platforms capable of level four (SAE J3016) autonomy require expensive sensing hardware (e.g. LIDAR), which limit the technology from being ubiquitous. We propose a monocular camera paired with a graphical computing unit (GPU) edge device [9] as the only hardware requirement. This makes the system architecture scalable towards scaled vehicle platforms, which can perform the diverse test requirements for robustness before full scale development. Furthermore, as a measure of safety standards, we utilize visual inspection of the neural network layers (e.g. Visualbackprop [4]) to provide the needed observability of the system model. With this hardware, we propose overtaking sequences as a means of collision avoidance. The benefit is that this approach combines local trajectory planning and collision avoidance as a single subsystem. This is a more natural solution as opposed to modeling collision avoidance and local path planning separately. There have been many separate attempts to solve collision avoidance and path planning, but efforts have not been made to solve them in an integrated manner. Thus, collision avoidance and local path planning is a coupled problem which must be solved conjointly, and overtaking maneuvers provide a solution.

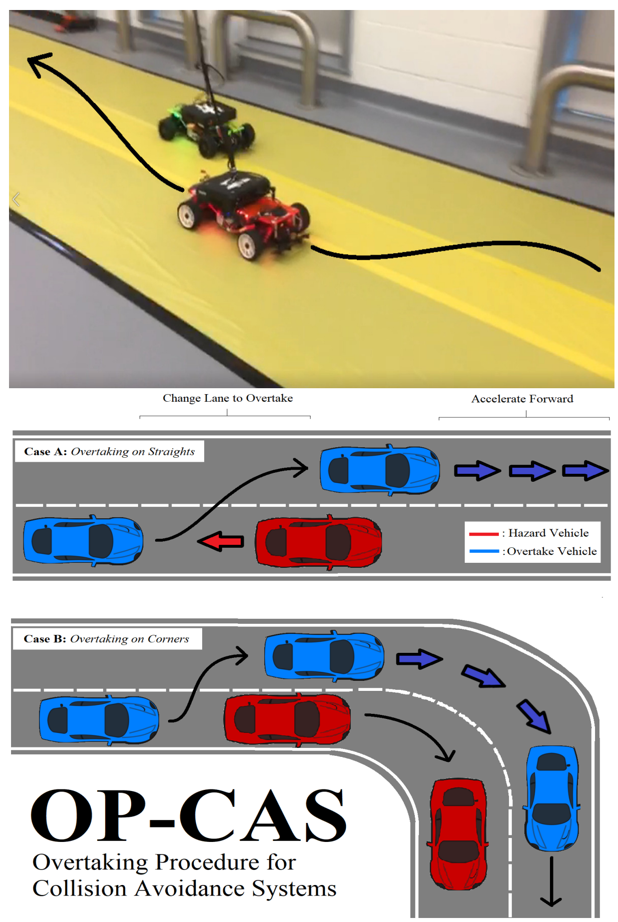

Fig. 1. ***OP-CAS***: **O**vertaking **P**rocedure **C**ollision **A**voidance **S**ystem
This collision avoidance system uses an overtaking procedure for various highway collision scenarios, as shown in Case A: "Overtaking on Straights" and Case B: "Overtaking on Corners"

Our paper presents an Overtaking Procedure Collision Avoidance System (OP-CAS) based on behavioral cloning.

This CAS system designed specifically for hazard highway scenarios uses overtaking procedures as part of a collision avoidance maneuver. OP-CAS is effective for two reasons. First, it integrates three of the four hierarchical layers of autonomous driving [3] (behavioral decision making, motion planning and vehicle control) as a joint problem when route planning is not necessary. This is advantageous to simple brake based CAS systems which eventually need a separate motion planner to resume driving. Second, a monocular camera is only needed for deployment, and can easily be complemented with additional vehicle state information. As a proof of concept, collision avoidance scenarios were evaluated with two scaled racing vehicles using the proposed method. Experiments showed that a dataset as little as 20 minutes was sufficient enough to capture the overtaking behavior. It is well known that behavioral cloning (BC) is susceptible towards covariance shift, which result in compounding trajectory error for the autonomous vehicle during operation. OP-CAS alleviates this issue by injecting additional ”lane-keeping” demonstrations (variant of DART [15]) into the supervisor policy during expert demonstrations of the overtaking procedure so that the vehicle avoids drifting. In conclusion, the contribution factors are as follows:

- *A novel, behavioral cloning based,* ***CAS architecture which uses recovery sequences to alleviate covariate shift****.*
- *A new method which* ***jointly addresses local path planning and collision avoidance through overtaking procedures****.*
- *The implementation of a scaled vehicle platform which can be used as* ***a scalable testbed for traffic optimization of autonomous vehicles.***

II. RELATED WORK

Model-Based Methods: Although there exist many model-based control methods [13] that attempt to address collision avoidance, they require expensive computational hardware for real time deployment. Most systems require knowledge of the vehicle state (e.g. localization), increasing sensor cost for collision avoidance maneuvers. [7] Furthermore, while model-based trajectory planning (e.g. Model Predictive Control (MPC) [13]) allows the observability often required with safety critical systems in transportation, non-linear mathematical modeling of the vehicle dynamics require simplification for real-time deployment at high speeds, compromising the robustness of the system. [6] [5]

Model-Free Methods: The idea of autonomous driving using end-to-end neural networks is not new and dates back to the 1980s [11]. Recently, significant progress has been made for this type of image-based autonomous vehicle control architecture. Unfortunately, it is widely known that vision-based control architectures alone cannot provide level five autonomy for civilian vehicles because it does not provide a means of navigation, although there have been efforts to alleviate this problem in recent years. [12] This being said, BC methods can be used for overtaking procedures during highway CAS scenarios where navigation is not necessary.

Critics argue that using neural networks for safety critical CAS is unsafe. The crux of the argument lies in the non-observable nature of end-to-end neural networks. However, the internal processing behind the neural network architecture can be analyzed through activations of its salient features [10] using techniques such as visual back-propagation. [4] Others argue that mimicking the behavior of other drivers cannot be based on a robust metric which guarantees a safe driving model, and lean towards control based methods which offer observability based on vehicular kinematics. We propose that the overtaking procedure can be based on data from professional drivers, whose driving behavior could be cloned to guarantee a high level of driving safety. Furthermore, this proposed BC solution can be used as pre-training with reinforcement learning to ensure optimality of the maneuver.

The main strength of this approach is that it only requires monocular vision for deployment in comparison to localization-based path-planning methods, and can easily be integrated with vehicle odometry from additional sensors to guarantee the required level of functional safety. (ISO26262, SAE J3092)

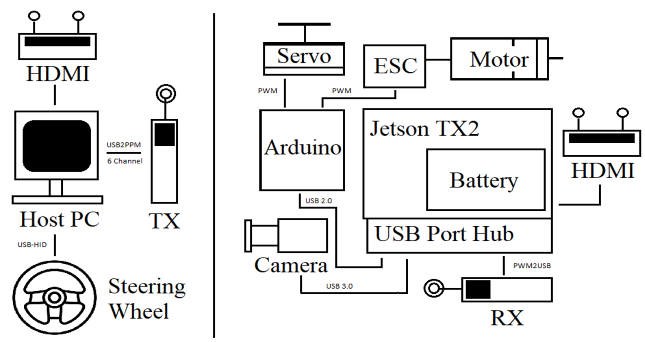

Fig. 2. Hardware System Architecture: An HDMI digital wireless transmitter minimizes latency with a gaming monitor at sub 11ms. An Nvidia Jetson TX2 board was used as the embedded processor. A CMOS camera with 185 degrees field of view (FOV) provided images at 80 frames-per-second (FPS).

III. METHODOLOGY

A. Assumptions

The assumption is made that the state information obtained from fish eye monocular images is sufficiently expressive in identifying the experts intentions when performing actions. However, an autonomous vehicle navigating towards different destinations within the same environment might look at the same state (e.g. intersection) and be forced to make different actions (left or right). For collision avoidance overtaking maneuvers, this assumption is overlooked as navigation is not needed for local trajectory planning. Also, our deterministic policy assumes that the expert is able to perform the task of driving well, although the optimality of this performance is not guaranteed.

B. Theoretical Objective

We use regression to model an approximator policy $\pi_{\hat{\theta}}$, which can approximate the expert policy, $\pi_{\theta}{}^*$ for the two

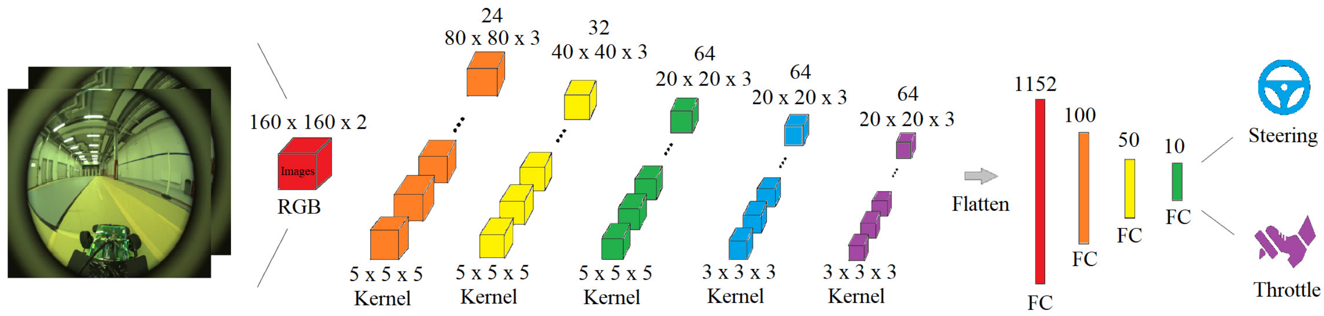

Fig. 3. Neural network architecture used for *OP-CAS*: The 3D-CNN model consists of an lambda layer to normalize the image pixels, 3 convolutional layers with 3 hidden layers with a kernel size of 5 x 5 x 5 and a stride of 1 x 2 x 2, then 2 additional hidden layers with a kernel size of 3 x 3 x 3 and a 1 x 1 x 1 stride. The channels are 24, 32, 64, 64, 64, 64, in descending order. The layers are flattened, passed through a fully connected layer of 1152 units, then converged into a final dense layer of 2 units, representing steering/throttle values. The network has a linear output activation function so the control values fall into a range of -1 to 1.

dimensional continuous action space $\mathbb{R}^2$. This policy $\pi_{\hat{\theta}} : \mathcal{S} \rightarrow \mathcal{A}$ maps the set of observable states $\mathcal{S}$ to a set of actions $\mathcal{A}$, using weights w of the neural network as to parametrize θ for the policy space Θ. Here, the trajectory τ of the vehicle consists of n number of state-action pairs $\{(x_1, a_1), (x_2, a_2), \cdots, (x_T, a_T)\}$ over time horizon T, whose transition probability is assumed to be Markovian. The probability density $p(\tau \mid \pi_{\theta^*})$ over the set of trajectories of length t can be expressed as $p(x_0)\Pi_{t=0}^{T-1}\pi_{\theta^*}(a_t \mid x_t)p(x_{t+1} \mid x_t, a_t)$, where $p(x_0)$ is the initial state distribution. The total cost $J(\theta, \theta^* \mid \tau)$, is the sum of the surrogate loss between the values of approximated steering $\hat{S}_t$, throttle $\hat{S}_t$ and the steering S_t^*, throttle T_t^* demonstrated by the expert when the vehicle traverses its trajectory. The steering and throttle inputs are unit-free and range between -1 to 1. The numbers directly represent the pulse-width-modulation signals coming from the Futaba Receiver.

$$\hat{\theta} = argmin_{\theta \in \Theta} \mathbb{E}_{p(\tau \mid \pi_{\theta^*})} J(\theta, \theta^* \mid \tau)$$

$$\text{Where,} \quad J(\theta, \theta^* \mid \tau) = \frac{1}{2N} \sum_{t=1}^{T} L(\hat{a}_t, a_t^*)$$

In short, we create an approximator policy $\pi_{\hat{\theta}}$, that can mimic the expert by minimizing the total cost i.e. cost-to-go using the Euclidean *L2* loss function to find the difference between the expert actions and predicted actions. The loss L can be expressed as follows:

$$L(\hat{a}_t, a_t^*) = L_{steering} + L_{throttle} =$$

$$\sqrt{\left\| \hat{S}_t - S_t^* \right\|_2^2} + \sqrt{\left\| \hat{T}_t - T_t^* \right\|_2^2}$$

C. Overcoming Covariate Shift

A problem with using a loss function as a deterministic policy is the assumption of covariate shift. [15] We assume that there is no discrepancy between previously labeled training images I^* from training set $(I^*, (S_t^*, T_t^*))$ and unseen, unlabeled test images $\hat{I}$. However, this assumption results in compounding errors [14] which make it hard for the approximator policy $\pi_{\hat{\theta}}$ to generalize the task of driving towards unseen states.Physically speaking, covariate shift makes the vehicle drift. The DAgger [14] method can be found to alleviate this issue via on-policy methods by providing corrections through explicit supervisor labels. This method was inapplicable for our scenario as the vehicles moved too fast for the expert to make adequate corrections real-time for a given overtaking maneuver. The method is also labor-intensive because it requires multiple iterations of training and testing. Our proposed OP-CAS algorithm instead uses a one-shot method for addressing this problem during training. This method injects ”return-to-lane” trajectories [15] [12] during expert demonstration.

Algorithm 1 *OP-CAS Algorithm*

Input: $\mathcal{D} \leftarrow \langle (x_t, y_t) : \forall n, \forall (x_t, y_t) \in \tau_n^*, \forall t \geq 0 \rangle$

inject $\tilde{\tau_n}$ *into* τ_n^*

Output: Corresponding action dataset $\mathcal{A}(\mathcal{D})$

$for\ t = 1 \ldots T\ do$ $\quad // train\ with\ regression$

$\quad x_t \leftarrow current\ observation$

$\quad \hat{a}_t \leftarrow \pi_{\hat{\theta}}(x_t)$ $\quad //query\ policy\ for\ action$

$\quad execute\ action\ \hat{a}_t$

$\quad L_t \leftarrow observe\ instantaneous\ loss$

$end\ for$

$return \sum_{t=1}^{T} L_t$ $\quad //return\ total\ loss$

As a solution, we present OP-CAS in Algorithm 1, which uses various overtaking maneuvers for collision avoidance scenarios. OP-CAS explicitly addresses the issue of the error term by concatenating an extra set of trajectories $\tilde{\tau_n}$ which demonstrate lane returning [12] as a variance reduction technique, so that even when the input images during testing differ from the training the approximator policy can quickly recover from covariate shift to avoid the accumulation of the error. The mechanism behind this noise injection is as follows:

1) Drive the vehicle in an oscillatory manner.
2) When the vehicle moves out of lane do not record; when the vehicle moves back into the lane, press a user-defined button to record the driving behavior.

Further details can be found in section four.

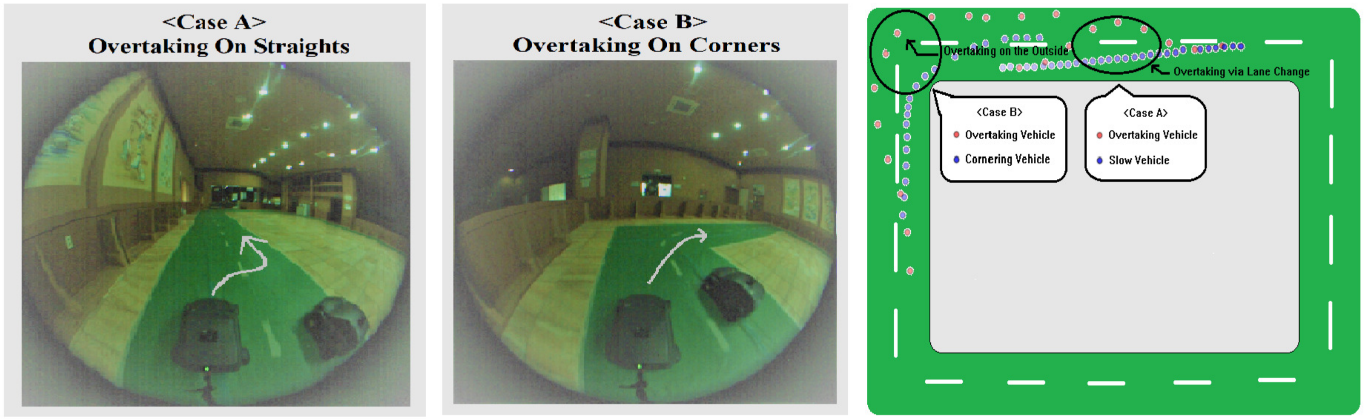

Fig. 4. Overtaking procedures used for *OP-CAS*: OP-CAS accounts for two cases of different overtaking maneuvers. The map on the right overlays the trajectory of the overtaking vehicle (red) and the vehicle that is being overtaken (blue).

D. Network Architecture

Three neural network architectures are investigated. The two dimensional convolutional neural network (2D-CNN) provides the initial baseline, which correlates steering/throttle values with a single image frame. Although it has the fastest deployment speed, it fails to account for temporal information of the vehicle state. The three dimensional convolutional neural network (3D-CNN) architecture attempts to incorporate temporal information by stacking two consecutive image frames for each control output. The method in which the images are stacked are described here. [2] Likewise, the Long-Term Recurrent Convolutional Network (LRCN) [1] architecture uses Long-Short Term Memory (LSTM) units to store previous state/action information as memory for the next control output. The 3D-CNN architecture outperforms the other two architectures and is selected as the OP-CAS algorithm. Exponential Linear Unit (ELU) activation function layers are used to express system non-linearity.

E. Salient Feature Representation

Salient features can be extracted from the convolutional neural network to represent which particular areas of the image most influences the algorithm's control decisions of steering and throttling. The features of an image which are most salient towards the control decisions of the OP-CAS algorithm are highlighted for observation and analysis. This is performed by using the activations of high-level convolutional feature maps as masks for the low-level convolutional feature maps. The salient feature activation mask uses a color scheme to determine which regions of the image are most important in the decision making process. As seen in the results, OP-CAS learns how to distinguish the vehicle from the road as well as the lanes and surrounding environment. [10]

IV. Data Collection

The dataset only consists of images correlated with the steering/throttle commands at each time step. Lane-keeping trajectories are injected into the BC model by collecting actions of the driver returning to its desired lane after an out-of-lane event. Oscillatory driving during this phase is divided in to large and small oscillations for robustness. The vehicle repeatedly drives out-of-lane, then returns-to-lane. The return-to-lane sequence is only recorded with a hot-key button. This requires quick hand-eye coordination in addition to driving ability, as the record button needs to be pressed at the precise moment for robust data collection. The vehicles are driven at various speeds to allow the model to fully understand the circuit. Optimal driving behavior is collected with the lane-keep assist trajectories to imitate the braking points and turn-in points of critical cornering events. The dataset uses both solo driving and two-vehicle driving to clearly separate lane keeping behavior with overtaking behavior.

- ***Case A**: "Overtaking on Straights"*
  One of the vehicles are controlled to perform a overtaking procedure on straights. This is commonly seen in cases of road obstruction. (e.g. animals, pedestrians) Other cases involve the vehicle ahead driving too slowly.
- ***Case B**: "Overtaking on Corners"*
  The overtaking vehicle takes the outside or inside lane to overtake the opposing vehicle.

160 x 160 pixel size RGB image frames are collected and down-sampled at 25Hz during dataset collection. Around 80,000 images are collected simultaneously from both vehicles for a 20 minute driving session. Half of the dataset contains 32 expert demonstrations of overtaking maneuvers and a quarter of the dataset contains the "lane-keep assistance" trajectories. Both vehicles were built identically to allow for transfer learning. Initially, the one vehicle is kept stationary at different straights and corners to test the robustness of OP-CAS. The second vehicle is autonomously driven at speeds slightly slower than the overtaking vehicle after the 40 stationary laps. Using these methods, two emergency scenarios are recreated in an attempt to validate the robustness of the proposed collision avoidance system.

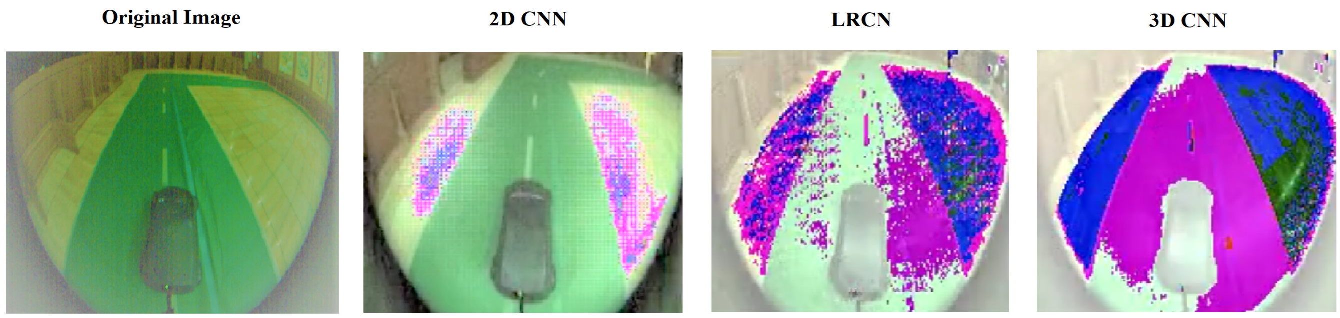

Fig. 5. Salient feature activations (purple) for different neural network architectures. Based on the amount of salient feature activations on the given image, 3D-CNN best distinguishes lanes from cars and from the road external environment.

V. Experimental Results

Experiments are performed to find the best architecture which can accurately predict throttle and steering values for a given image. Results show that out of three tested architectures, (2D-CNN, 3D-CNN, LRCN) the 3D-CNN model shows the greatest promise in accurately predicting both steering and throttle values for overtaking maneuvers and lane-keeping. Experimental results with the lane-keep assist trajectory injection show that the vehicles perform substantially better with an increased number of lane-keep assist demonstrations. Results also show greater stability for the overtaking maneuver with an increase in the number of overtaking demonstrations injected to the dataset. Doubling the number of overtaking demonstrations from 12 to 24 increased the accuracy from 50 percent to 75 percent. This is promising because it indicates that an increase in data during training can improve the performance of the OP-CAS algorithm. Table 1 shows the different neural network architectures measured against each other using R-Squared accuracy as a metric. Comparison between different architectures are also visually inspected by looking at the salient feature activations of each respective model as shown in Fig. 5. 3D-CNN showed the greatest amount of activations for a given image and was selected as the OP-CAS algorithm. A radar speedometer gun is used to measure the top speed of the overtaking vehicle during the overtaking maneuver. The positions of the vehicles during overtaking procedures are captured with ultrasonic indoor positioning beacons (MarvelMind) and can be seen plotted in Fig. 4. The results are promising with the faster vehicle successfully overtaking the slower vehicle, but further work is necessary as the overtaking vehicle sometimes fail to stay within lanes after an overtaking event during corners.

TABLE I
R-Squared Accuracy of Network Architectures for OP-CAS

Image Size: 157 x157 Pixels RGB		
Type of Architecture	***Steering (%)***	***Throttle (%)***
2D-CNN, 1 Stacked	93	66
3D-CNN, 3 Stacked	98	92
LRCN, 3 Stacked	89	38

TABLE II
Vehicle-to-Vehicle, 20 Autonomous Laps

Table Column Head		
Driving Scenario	***Case A***	***Case B***
# of Successful Overtakes	19 out of 20	16 out of 20
# of Collisions during Overtaking	1	4
Top Speed during Overtaking	24 (Km/h)	24 (Km/h)

VI. Discussion

The OP-CAS algorithm is able to be functionally implemented on the RC vehicle for overtaking maneuvers. The vehicles perform overtaking maneuvers for both corners and straights with over 85 percent accuracy during the experiments using just 20 minutes of data. OP-CAS proves to be more robust during straights, where it only fails to overtake once out of twenty attempts. Also, the vehicles are usually able to keep within the lane before overtaking. Although we propose OP-CAS, it may be safer to just stop the vehicle, especially if the preceding vehicle exhibits erratic behavior. (e.g. driving under the influence.) Also, the importance of safety must not be downplayed. Therefore, the assumptions and limiting factors in which the OP-CAS algorithm was validated will be thoroughly discussed.

First, the OP-CAS algorithm is only verified under the assumption that the preceding vehicle displays predictive driving behavior, meaning it keeps within lanes at all times at constant velocity. (e.g. no emergency braking or lane changing) Second, only two scaled vehicles are tested on a simplified rectangular circuit, with both vehicles traveling in the same direction and a lane always available for use during overtaking. OP-CAS would need further improvement for multiple vehicle scenarios. Third, only 20 minutes of training data is implemented with only 20 test runs to validate the system due to battery constraints. Fourth, OP-CAS requires an expert driver dexterous enough to perform oscillatory driving and selective driver behavior recording to also demonstrate robust driving. Fifth, it requires the driver to demonstrate lane keeping behavior at high speeds for diverse situations. Last but not least, this validation was done on scaled vehicle platforms, which may not represent full scale vehicle dynamics.

If OP-CAS were to be validated for real world applications, a more thorough validation study would have to involve real vehicles being tested with larger batches of training data on diverse environmental conditions (e.g. snow, glare, opposing vehicles) for the algorithm to be robust for civilian automotive applications.

In a high speed environment, the state of the car has temporal features during acceleration, braking or even steering. The 2D-CNN algorithm does not account for this temporal information because it acquires single static images as input and does not use a recurrent neural network as time- distributed memory. The LRCN model also has considerably more activated pixels when applying the deconvolution for the salient features. However, the LRCN model does not outperform the OP-CAS model. The 3D-CNN architecture used for OP-CAS proved to perform reasonably well at predicting the throttle input actions from the training data as seen in Table 1.

The computational power of the embedded processor restricts how fast it takes for the model to predict action commands. Due to these limitations, once the vehicle reaches beyond its top speed of 24 km/h, the vehicle has trouble exhibiting robust driving behavior. The challenge of oscillatory driving at high speeds is met by separating the training phase for lane departure and overtaking maneuvers. Second, the oscillatory driving required for learning lane departures is kept to a minimum and performed mostly at low speeds.

The salient features of the three algorithms are visualized in purple and in blue. The purple and blue salient feature masks are the areas that the OP-CAS algorithm uses to output action commands. It should be noted that OP-CAS only considers the surface of the floor as relevant; it does not show activations both on the environment or on the vehicle. Furthermore, the purple and blue colors clearly show that the algorithm is able to distinguish the split lanes and the surfaces which do not belong to the road lanes. The blue color indicates that OP-CAS uses more lower-level and higher-level convolutional feature maps for the split lanes and off-road surfaces when compared to the purple activations.

VII. Conclusion

We propose OP-CAS, a real-time collision avoidance system using overtaking maneuvers. This research has a broader reader impact in that it outlines a method of alleviating covariate shift for behavioral cloning, which may be utilized for not just driver behavior modeling, but also other behavior prediction tasks. For example, this method can apply towards prediction tasks such as walking or arm manipulation. If we know how the system will behave in a failure event; e.g. tripping or dropping an object, an early recovery sequence can be augmented into the dataset just like the proposed OP-CAS algorithm. Furthermore, this work also details the build of a scaled vehicle research platform which could be used to test a diverse range of ADAS algorithms. Finally, further work is necessary with more data collected for training, also incorporating scenarios involving more diverse traffic conditions. As future work, OP-CAS may also be incorporated with a Unity simulator for trajectory optimization using reinforcement learning for future work.

VIII. Acknowledgment

This work was supported by the Advanced Vehicle Engineering Center (AVEC) at Cranfield University, the Artificial Intelligence & Probabilistic Reasoning (AIPR) laboratory at the Korean Advanced Institute of Science and Technology, and the Laboratory for Embedded Machines and Ubiquitous Robotics (LEMUR) at the University of California, Los Angeles. Special thanks to Haanvid Lee, David Kim from the Department of Computer Sciences at KAIST, Dr. Kim Blackburn from AVEC at Cranfield University and Nikita Ermonlenko for technical advice. Finally, I thank Nahee Lee for her support.

IX. References

[1] Jeff Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko and Trevor Darrell. "Long-term Recurrent Convolutional Networks for Visual Recognition and Description.", Conference on Computer Vision and Pattern Recognition (CVPR), 2015.

[2] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, Manohar Paluri, Facebook AI Research, Darthmouth College. "Learning Spatiotemporal Features with 3D Convolutional Networks.", International Conference on Computer Vision (ICCV), 2015.

[3] Brian Paden, Michal Cap, Sze Zheng Yong, Dmitry Yershov, Emilio Frazzoli. "A Survey of Motion Planning and Control Techniques for Self-driving Urban Vehicles", ArXiv, cs.RO 1604.07446v1, 2016.

[4] Mariusz Bojarksi, Anna Choromanska, Krzysztof Choromanski, Bernhard Firner, Larry Jackel, Urs Muller, Karol Zieba. "VisualBackProp: efficient visualization of CNNs", ArXiv, cs.CV 1611.05418v3, 2017.

[5] Xiaohui Li, Zhenping Sun, Dongpu Cao, Daxue Liu, Hangen He, "Development of a new integrated local trajectory planning and tracking control framework for autonomous ground vehicles", Mechanical Systems and Signal Processing (MSSP), 2015.

[6] R.S. Sharp, Huei Peng, "Vehicle Dynamics Applications of Optimal Control Theory", Vehicle System Dynamics, International Journal of Vehicle Mechanics and Mobility (IJVMM), 2011.

[7] Xiaohui Li, Zenping Sun, Dongpu Cao, Zhen He, Qi Zhu, "Real-time Trajectory Planning for Autonomous Urban Driving: Framework, Algorithms and Verifications", IEEE/ASME Transactions on Mechatronics, 2015.

[8] Liang Li, Guangquan Lu, Yunpeng Wang, Daxin Tian, "A Rear-End Collision Avoidance System of Connected Vehicles", IEEE International Conference on Intelligent Transportation Systems (ITSC), 2014.

[9] M. Bojarksi, et. al, "End to End Learning for Self-Driving Cars", ArXiv: cs.CV 1604.07316, 2016.

[10] M. Bojarski, et. al, "Explaining How a Deep Neural Network Trained with End-to-End Learning Steers a Car", ArXiv, cs.CV 1704.07911, 2017.

[11] D. Pomerleau "ALVINN: An Autonomous land Vehicle in a Neural Network", Conference on Neural Information Processing Systems (NIPS), 1988.

[12] F. Codevilla, M. Muller, A. Lopez, V. Koltun, A. Dosoviskiy, "End-to-End Driving via Conditional Imitation Learning", International Conference on Robotics and Automation (ICRA), 2018.

[13] J. Ji, A. Khajepour, W. W. Melek, Y. Huang, "Path Planning and Tracking for Vehicle Collision Avoidance Based on Model Predictive Control with Multi-constraints", IEEE Transactions on Vehicular Technology, 2017.

[14] S. Ross, G. J. Gordon, and J.A. Bagnell. "A reduction of imitation learning and structured prediction to no-regret online learning", International Conference on Artificial Intelligence and Statistics (AISTATS), 2011.

[15] M. Laskey, A. Dragan, J. Lee, K. Goldberg, and R. Fox. "Dart: Optimizing Noise Rejection in Imitation Learning", Conference on Robot Learning (CoRL), 2017.

[16] M. Kääriäinen. Lower Bounds for Reductions", Atomic Learning Workshop, 2006.

[17] Joseph Funke, Matthew Brown, Stephen M. Erlien, and J.Christian Gerdes. "Collision Avoidance and Stabilization for Autonomous Vehicles in Emergency Scenarios.", IEEE Transactions on Control Systems Technology, 2017.ystems Technology, 2017.