Aggressive Driver Behavior Detection using Parallel Convolutional Neural Networks on Simulated and Real Driving Data

Zehra Camlica, Jim Quesenberry†, Daniel Carballo†, Mark Crowley‡
‡*Dept of Electrical and Computer Engineering*
University of Waterloo
Waterloo, Canada
{zcamlica, mcrowley}@uwaterloo.ca

†*Corporate R&D*
Magna Int.
Troy, MI, USA
{jim.quesenberry, daniel.carballo}@magna.com

***Abstract*—The novel method proposed in this paper is compromised of application of two Convolutional Neural Networks (CNN) working in parallel to simultaneously classify driver behaviors while classifying maneuvers by using time series data. We claim that the Parallel Convolutional Neural Network (PCNN) not only speeds-up training time but also increases performance since having information about the maneuver helps to improve behavior classification performance and vice versa. In this study, both simulation and real-world driving datasets are utilized for driver behavior analysis. As simulation data, mobile phone sensor data are simulated as a time series using a combination of a traffic simulator (SUMO) and a car simulation system (Webots). The same type of data is collected with a specially designed vehicle traveled on a defined route around a predefined region. The collected data are then separately utilized as training and testing data for classification of both maneuvers (e.g turns and lane changes) and driver behaviors (e.g aggressive, non-aggressive) applying a novel method using deep learning on time series data. In addition, other methods which are commonly used for time series analysis, Hidden Markov Models(HMMs) and Recurrent Neural Networks (RNN), are applied to the same datasets to compare with PCNN. According to the results, the CNN classifiers perform efficiently for a single task and PCNN outperforms both single task-CNN and RNN with an average accuracy of 86%.**

I. Introduction

There are three major causes to road traffic accidents: human error, vehicle failure and road conditions. The NHTSA lists speeding, following improperly, erratic lane changing, passing where prohibited as the leading causes of driving fatalities due to aggressive driving. Just speeding is responsible for 11,258 fatal crashes on the roads in 2020 [1]. Perhaps not surprisingly, the first cause is the most influential [2] so understanding driver behavior is key to improving road safety. Furthermore, driving behavior can be positively influenced when a driver is monitored and behaviors are recorded [3]. There is already a strong history of driver behavior detection systems that try to ensure both driver safety and compliance to driving regulations [4]. More recently, Advanced Driver Assistance Systems (ADAS) have utilized classification and pattern recognition approaches [5]. **In this paper, we present a novel method for driving behavior analysis, which is our Parallel Convolutional Neural Network as a classification method.**

Driver behavior is a generic term including different driving maneuvers and driving mannerisms consisting of many variables. Driver behavior analysis can be decomposed into two complementary sub-problems: **driving maneuvers detection** and **driving characteristics analysis**. For instance, driver maneuvers can include turns, lane changes, stops, accelerating and decelerating events. There are many different types of driver characteristics in the literature such as speeding, distracted driving, fatigued driving, inattentive, aggressive driving and drunk driving. In this paper the following are studied: Normal Driving and Aggressive Driving for turns at intersection and lane changes.

Driving maneuvers are parts of a trip consisting of characteristics of driving patterns. The common driving maneuvers types include: stopping, acceleration/deceleration, lane changes and intersection turns, driving around curved roads and roundabouts, entry and exit to highway ramps. In order to derive these manoeuvre types, each maneuver should be analyzed according to vehicle kinematics. For example, lane changes and intersection turns are inferred from angular velocity and lateral acceleration patterns [6]. Driver behavior applications aim to detect **driving characteristics** which are out of the ordinary or abnormal [7]. One way to achieve this is to first defining normal or abnormal driving characteristics and then deriving different styles by calculating deviations from these. For this reason, some machine learning studies have just considered training individual driving history rather than using general rules or thresholds when determining style of driving [8]. Different data sources can be used to detect driver actions and driving characteristics: in-vehicle and smartphone. With the rapid development of today's information and communications technology, mobile devices such as smartphones are capable of collecting spatio-temporal information in real-time with onboard sensors [9]. Reducing the overhead costs in designing and implementing a sustainable large-scale are the main advantages of analyzing driver behavior via smartphone-based systems.

II. Related Work

Recently many projects have conducted research on driving behavior analysis. Different machine learning algorithms have been applied to learn and model driver's behaviors such as: Artificial Neural Networks (ANN), Support Vector Machines (SVM), Fuzzy Logic (FL), Dynamic Bayesian Networks (DBN), Random Forest (RF), Convolutional neural network (CNN), and Hidden Markov Models (HMM). [3] and [10] are review papers that evaluate recent studies which is on the driver behavior characteristics analysis. Artificial Neural Networks (ANNs) have increasingly gained more interest in the field of driver behavior and maneuver detection [11]. [12],[13], and [14] are some examples of Artificial Neural Networks applications based on computer vision techniques in drowsiness detection, distraction detection and steering behavior prediction respectively. ANNs can be applied on human behavior modeling since human behaviors can successfully be represented by using nonlinear and stochastic models. In the paper [11], ANNs were used to classify aggressive, normal and calm driver behaviors. The dataset consisted of acceleration, speed and throttle features which were been collected from a simulator.

Other approaches [15], [16] have used clustering techniques to detect and label driving styles. Wang et al. [15] proposed a semi-supervised SVM method to classify aggressive and normal driving style. HMM are also a commonly used method to analyze driver behavior [3] especially to estimate driving behavior [17] and to recognize driving maneuvers [18]. The study [19] uses HMMs in order to characterize and detect aggressive driving maneuvers for calculation of car insurance fee through estimating the driver aggressiveness, providing a summary of existing driving style studies with HMM method. Cervantes-Villanueva et al. [20] applied RF, SVM, and fuzzy rule-based classifiers for detecting maneuvers which are stopped, driving, parking, and parked. They used smartphone collected accelerometer data in this study and the RF results are the best in the two-level of classifiers. In [21], MLP, SVM, RF, BN are applied to classify driving maneuvers on smartphone sensors' data. The paper also compares these methods and combinations of the supervised machine learning algorithms for classification of seven driving maneuvers.

III. Our Approach

The main components of the aggressiveness driver analysis system are a map, a traffic simulation, a controller and a deep classification tool (see Fig 1). For the purpose of learning to identify aggressiveness in driving, we first started to train and test the proposed model in this study on a **simulated dataset** and on a newly collected real-world driving resulting in the **Driver Behaviour Learning (DBL)** dataset.

A. Data Simulation

Data simulators can provide a low-cost solution to modeling a wide range of scenarios in a realistic manner in order to predict the consequences of urban infrastructure modifications. A microscopic traffic simulator, called Simulation and Urban Mobility (SUMO) and Webots Simulator, were chosen as initial step for this study since they allow us to model vehicles individually and to control their features including driving behaviors(i.e. aggressive and normal).

The first phase of data simulation is data generatione which proceeds as follows: **(1)**The Open Street Map (OSM) is downloaded online by selecting random latitude and longitude from the world city latitude and longitude list. **(2)** A road network is created from OSM in order to be utilized for SUMO. **(3)**The population of vehicle agents are defined randomly from a defined range of sizes. Different vehicle types are defined for different driving behaviors (aggressive and normal) by changing parameters based on previous works parameters[22][23]. **(4)**The probability of these vehicle types are distributed randomly. **(5)**Random trips are created by selecting random two nodes for start and end points for and by utilizing numbers and distribution of vehicles.

The second phase is the application all information which are random trips, vehicle types, and numbers into the simulator: **(A)** SUMO utilizes network, trips and vehicle types information to simulate traffic and save data. **(B)** Labels can be created successfully according to output data of simulation.

1) SUMO Simulator: In order to simulate driver behaviors, we need to not only simulate the traffic but also allow control of the vehicle's features and behaviors. SUMO [24] is an open source multi-modal traffic simulation tool that can simulate multiple types of vehicles. It is a type of "microscopic" simulator, meaning that each vehicle moves its own route, and behaves individually. SUMO has a continuous space, and it records data for each vehicle in discrete time (the default time step is one second) and can include traffic lights. We have used SUMO version 0.12.3 2 in this study. The vehicle dynamics of driving on the road in SUMO are determined by several models, the most important of which for driver behaviors are: **car-following model** that controls a vehicle speed according to the vehicle ahead of it, **intersection model** that controls vehicles decision at different types of intersections based on right-of-way rules and gap acceptance and **lane-changing model** which determines lane position on multi-lane roads and adjusts speed for lane changing.

2) Webots Simulator: Webots is an efficient open source robotics simulation software using the Open Dynamics Engine (ODE) library. Webots provides a real-car based vehicle model with realistic dynamics features with ODE. Also, multiple intelligent vehicles can be simulated in Webots. In Webots simulation world, the Open Street Map (OSM) data is used to extract the road network and its features. The OSM maps are converted to a graph data structure for Webots. SUMO simulates traffic and controls each vehicle behaviors, while Webots simulates each vehicle dynamics, enables to add sensors into vehicles and calculates outputs of these sensors based on coming traffic information from SUMO.

B. Real-World Driving Behaviour Learning Dataset (DBL)

Our second dataset comes from a study we conducted over 2019-2022 to observe human driving behavior in real-world conditions across a range of road types, driver age groups and experiences. The study gathered 50 volunteer participants who drove a predetermined route (See Fig. 3) around the region of Waterloo, Ontario, Canada. Due to the COVID-19 pandemic most drivers ended up driving the route entirely on their own, following direction from an on-board GPS navigation system. The vehicle for the study, was equipped with a top-mounted Lidar, front and two rear-facing radar, camera sensors, onboard

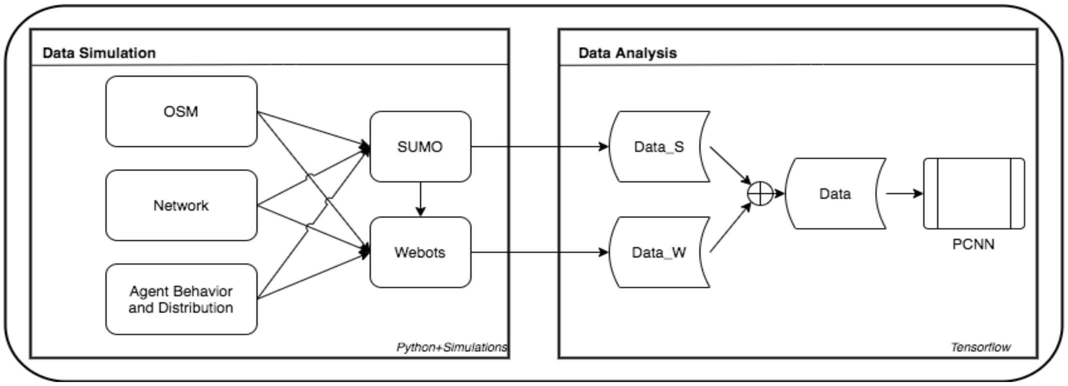

Fig. 1: General system architecture for the driver simulation tool.

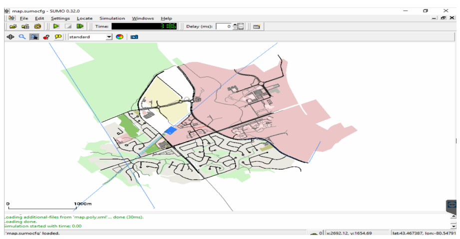

Fig. 2: An illustration of the GUI for the SUMO traffic simulator.

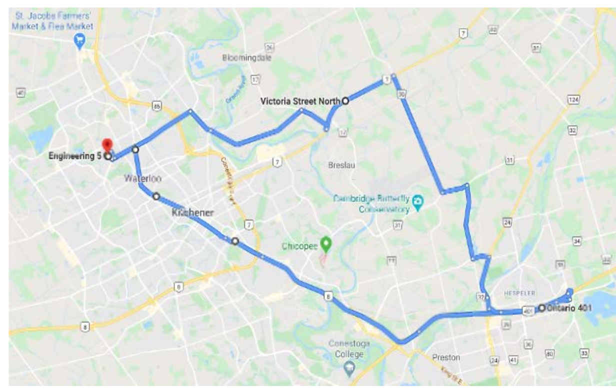

Fig. 3: The route that used for participant drives

computers, and various readout displays for the passenger. The onboard computers collected GPS, map, Lidar, Video System (VLT) and CAN-BUS data. A self-evaluation with an exit survey and annotation data from research staff was utilized to create the Driver Behaviour Learning (DBL) dataset used in this paper.

Data was collected on multiple road types (e.g. major highways, city centers and rural highways), under different traffic conditions (e.g. rush hour, traffic congestion times, low traffic) and under different environmental conditions (winter, spring, summer). The same route was repeated twice for each participant, with a short break in-between. The route length is around 57 km which takes about 1 hour 10 min to complete without traffic. One tour consists of approximately 12 Right turns, 11 Left turns, one forced left lane change, three forced right lane changes and two straight roundabouts.

1) Route: City roads (urban environment) trips are compromised of city-specific features such as: pedestrians, crosswalks, traffic lights, stop signs, cyclists, roads affected by building constructions, parked-cars, turns, narrow roads, etc. The highway parts consists of features such as: curved roads, multi-lane sections, on and off-ramps, connections between highways via on-ramps, varying speed-limits, and highway divisions. The country roads includes features such as: stop signs, yield signs, roundabouts, traffic lights, single-lane two-way roads, turns, varying speed limits, traffic, etc.

2) Participants and Procedure: All participants were selected as experienced drivers with full 'G' licenses to drive in Ontario on all road types. Participants were prioritized based on the diversity of the applicant's information to ensure that an appropriate cross-section of the population were selected. An annotator sat beside the participant for the first 24 participant drives. The first 20 participants' drives are labeled and utilized for driver behavior analysis in this study.

Research Ethics Board Approved: The DBL dataset was collected over the years 2019-2022 during a human participant study (*"Driver Behavior Learning [REB: 31381]"*) which underwent rigorous approval with the University of Waterloo Office of Research Ethics to ensure that the safety, privacy and health of the driving participants was protected at all times.

IV. Parallel Convolutional Neural Networks

Convolutional neural networks consist of alternating convolution and pooling layers which allow more efficient learning of high dimensional data than a fully connected network by leveraging weight sharing and a locality assumption. The motivation for our research is to use the efficient weight sharing and locality assumptions of CNNs to develop a richer model for multidimensional time series data.

To utilize CNNs, we first need to transform our raw sensor 1D time series into a meaningful 2D form. We create 2D image representation of the data by using each feature such

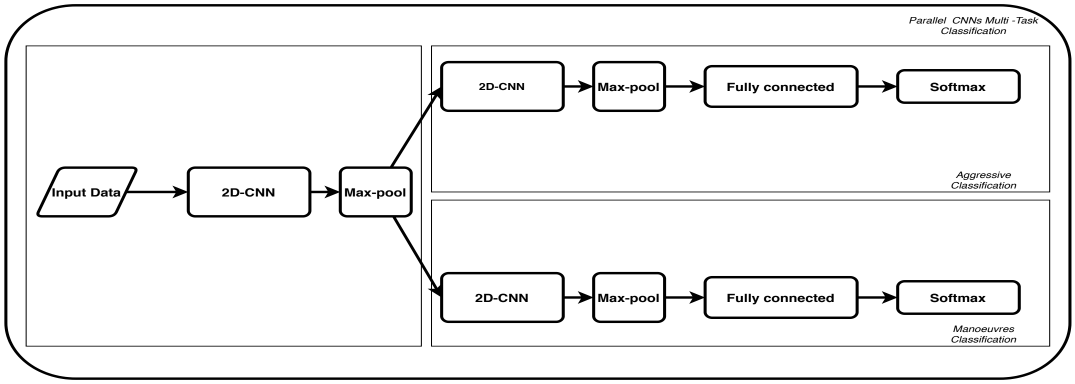

Fig. 4: Parallel CNN architecture for two different tasks, driving maneuvers and driver behaviors.

as position, velocity and angular velocity to define rows of an image. We then generate the columns of our data using a sliding window across time on the original time series data. Following the windowing, the data is of dimension `windowsize×featuredimension`. This dataset is then input into the CNN for training.

For classification sliding window sizes of 200 and 300 steps are applied with step size of 2 on position, velocity and angular velocity data derived from SUMO **simulated data**. Each window is labelled based on consistency of an event. Windows without any labelled events are labelled as 'other'. Turn events crossing window divisions were separated from the dataset and utilized for turn classification. The whole dataset consists of 163,089 events (turns and lane changes) in total. To achieve a class-balanced, the class with the fewest samples determined the train/test data set sizes (i.e. Aggressive Rlch has 7,625 samples). For turn event classification, 15,250 events, equally distributed between aggressive and normal, were used with 80% used for training and 20% for testing. Also, lane changes were separated from the dataset and utilized for lane changes classification. There are 15,250 lane changes which have been selected from the entire dataset consisting half of them aggressive and same percentages (80%-20%) of lane changes events used for training and testing.

As with the simulated data, sliding window sizes of 2-3 seconds are applied with overlapping 2 step size on position, velocity, and angular velocity data coming from **DBL**. If the sample window has both maneuver and behavior labels, it is selected as the main dataset to be used for the proposed classification method. The whole dataset has more than 120,000 sample event windows. However, in order to balance the datasets for each class, downsampling is applied and the aggressive left lane change class sample size is selected as the maximum number for each class which is 3,904. Samples from other maneuver classes are selected randomly based on this sample size. 15,616 events combination have been generated for 10 runs including 7,808 aggressive actions. 70%, 10% and 20% are the percentages that are used for training, validation and testing respectively.

A. *Parallel Convolutional Neural Network Implementation*

The Parallel CNN architecture (in Figure 4) we build for the aggressive driver identification problem consists of two CNN classifiers being trained and working in parallel as follows. The first Convolution and Pooling layers are used for both classification tasks in order to share weight and parameters since we propose that these two classification tasks are related and can support each other by sharing. The remaining convolution and pooling layers and fully-connected layers are separated to focus on their specific purposes for the classification problem. By applying this method, efficiency for both results and time is improved.

In order to compare with PCNN, we have implemented LSTM and HMM methods by using same plain time series dataset and features. For HMM, each class from 8 classes has own HMM. We trained 8 separate HMM models and the HMM classifier is working based on winning log-likelihood output from HMMs. Different numbers of hidden states have been tried and three states are selected for this problem. Moreover, the same dataset have been utilized to run LSTM consisting biLSTM, fully connected and softmax layers. Adam is selected as the optimizer algorithm and the data is shuffled for every epoch.

V. Experiments

In this work, we study an important real-world problem of driver behavior analysis and propose a deep learning based solution. Initially, our proposed approach is validated on simulations, a micro-traffic simulator SUMO and robotic simulator Webots. Then the method is also applied to the DBL dataset.

During simulation tools, many trips were recorded with different parameters which control aggressive driver behaviors. A range of window sizes were used to extract a time series data points using a sliding window approach. After windowing, the trips are labelled automatically according to simulator features in terms of behavior (normal/safe, aggressive) and manoeuvre (right turn, left turn, right lane change, left lane change). Aggressive vs normal driver behaviors are simulated by tuning the SUMO parameters I based on documentation and previous

Parameters	Normal Driver behavior	Aggressive Driver behavior
acc(m/s^2)	2.6	3.2
decel(m/s^2)	4.5	5.5
line sigma$(0-1)$	0.5	0.3
maxSpeed(m/s)	23	35
minGap(m)	2.5	0.9
tau (s)	1.5	1
impatience$(0-1)$	0	1
lcAssertive$(0-1)$	0	0.9

TABLE I: SUMO Parameters for aggressive and normal driver behaviors

studies [22][23][25]. Each driver type will be assigned a specific value for minimum gap acceptance, reaction time, acceleration and deceleration rates, maximum speed and etc. Simulation data are generated according to randomly selected region road map information, random vehicle distributions and determined driver and vehicle parameters. The output of simulation includes many features, but only position, velocity and angle information have been utilized to apply the proposed method, some samples of data are shown in the Figure 7. This time series dataset are used as **plain, rotated** formats which are:

- **Plain Time Series without Reformatting:** 2D Data output taken from directly simulators. Dimensions are time in time window (300ms) and vehicle features (speed, x, y, angle) from simulations (4x300 for just one maneuver sample).
- **Time Series with Rotated Location x and y:** Zero centered x and y position data has high spatial variance within single class. In order to translate this into more informative format for any application method, we rotated each (x,y) datapoint so that them all to be in the first and second quadrants of the x and y coordinate plane. Essentially we normalized the initial direction for each maneuver to North. Figure 6 illustrates positions before and after rotation application.

The proposed model was then applied to the selected 20 participants from DBL dataset. Position, velocity and angle information that are coming from data are re-sampled and labeled manually by watching driving Go-Pro video by a researcher of the project. We have **determination rules for aggressiveness** which are defined by previous studies[26],[27]. These rules which we considered to label the action as aggressive are speeding, accelerating and decelerating quickly, close following, frequent and improper lane changing, failure to signal and failure to obey traffic signals, racing, and frequently honking. For each action, start points are labeled manually by using Matlab video labeler tool shown in Figure 5. Each participant's video is watched and labeled by the researcher and controlled these labels after each labeling process is done. Then, the annotator's notes were checked before and after the labeling process in order to catch some aggressive drivers' behaviors that can not be seen from the video such as shouting or honking. After visually labeling, the actions

Aggresiveness rules	Threshold
acc/decc quickly(m/s^2)	$\|ax\|>3$
turn quickly(rad/s)	$\|gyroz\| >1.2$
speeding(km/h)	$>$ speed limit+25
close following (m)	stand still distance<0.4
close following(sec)	head way time<0.5
frequent lane change$(/min)$	total numbers of lane change>8

TABLE II: Aggressive driver behaviors determination rules and thresholds

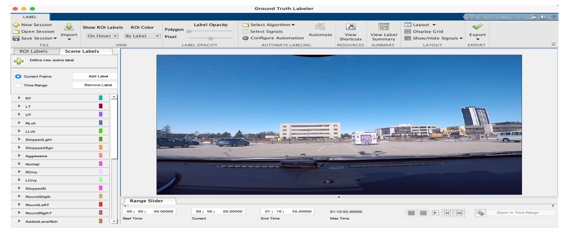

Fig. 5: Manual labeling tool

are also controlled and labeled by automatically considering aggressiveness rules which are defined numerically in Table II. The same range of window sizes was used to extract a time series of data points using a sliding window approach. After windowing, these windows are labeled according to consisting of aggressive behavior (normal/safe, aggressive) or maneuver labels (right turn, left turn, right lane change, left lane change) within these time windows.

A. Results

We first implemented four separate 2D CNN models (illustrated in Figure 8) for different tasks to apply on time series data for both simulated and real-world driven data. Turn classification results with a single task CNN as shown in Table III are 90.2% and 80.3% for training and testing on the simulated dataset. Lane changing classification with the single task CNN gives a little less than turning but still has captured significant features from the simulated dataset to classify. When these turn and lane-change events are all combined into *maneuvers* for another single task CNN classification, the accuracy decreases to 82.9% and 65.0% for training and testing on the simulated dataset. Since more classes increase the complexity of the problem, the combination of lane-changing and turn classification accuracy is lower than only lane-changes or turn classification accuracy. Aggressive and safe behavior classification performance are

Classification Task	Train accuracy (%) on Simulated Dataset	Test accuracy(%) on Simulated Dataset	Train accuracy(%) on Real Driven Dataset	Test accuracy(%) on Real Driven Dataset
Turns Classification	90.2	80.3	92.1	90.5
Lane-changes Classification	88.2	75.6	80.6	72.7
Behavior Classification	65.6	60.8	80.5	78.4
Manoeuvre Classification	82.9	65.0	75.3	71.1
Behavior Classification with Manoeuvres label added to input features	73.1	65.0	88.2	82.7
Manoeuvre Classification with Behavior label added to input features	89.2	70.3	78.2	75

TABLE III: Accuracy results for different tasks using just the **single task CNN** on time series datasets.

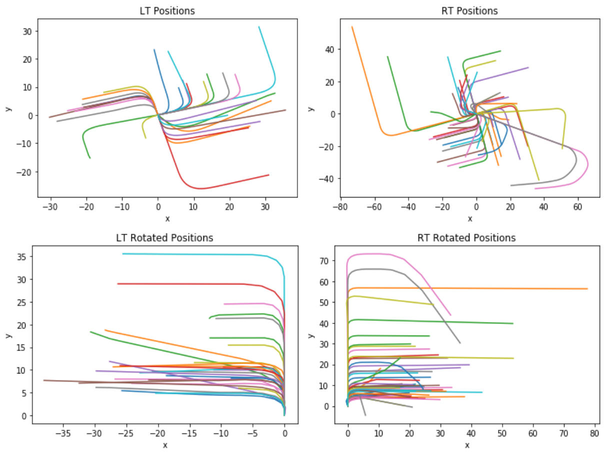

Fig. 6: Zero centered vehicle x and y position data samples for 300 time window from the Dataset: Left side plotting shows left turning maneuvers' x and y location data from Simulator, right side plotting illustrates right turning maneuvers' positions. Rotated vehicle x and y position data samples for 300 time window from the Simulated Dataset

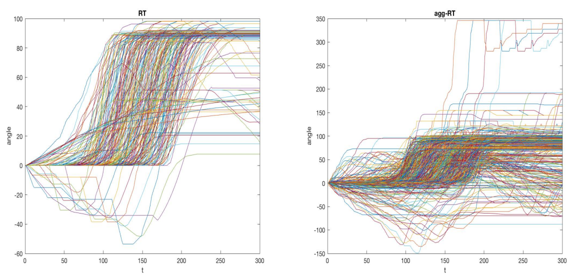

Fig. 7: Vehicle angle of Right turn sequences and Aggressive Right turn sequences from the Simulated Dataset: Plots shows time vs angle data for right turning maneuvers. In aggressive right turn, most of the angles are more sharply changing and waving and also event time is smaller than the non-aggressive one in general.

65.6% in training and 60.8% in testing because this CNN classification is more challenging task compared to others. When we include all 8 classes together, this classification task even becomes even more complex and the accuracy decrease to 70% and 61%, training and testing respectively, see Table IV. When single task CNN models are fed with labels of other classification classes, classification accuracies rise up. For example, if aggressiveness classification samples include maneuver labels, the classification efficiency shows a remarkable increase from 80.5% to 88.2% in training accuracy on the real-world driven dataset. The results illustrate that the applied CNN classifications can improve their performance by adding other task labels as a feature set on input. We can infer from these results, PCNN can serve our purpose which is to show that high level of features derived from different tasks can support each other.

The PCNN model has been applied on **simulated time series** in order to classify these 8 classes. PCNN has improved accuracy results from 60.6% (single CNN model accuracy result) to 65.7% which suggests that sharing the early layers

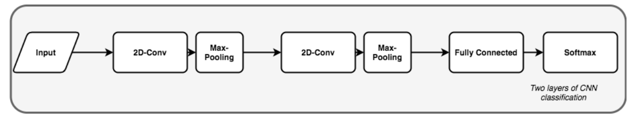

Fig. 8: 2D CNN model for single task

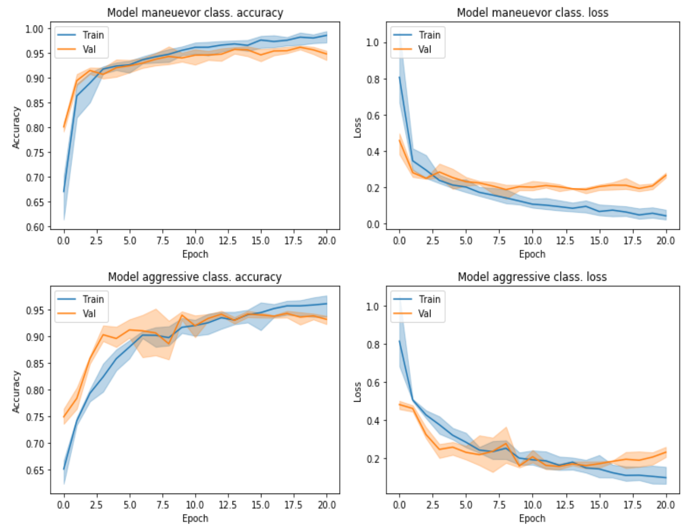

Fig. 9: PCNN classification on **rotated time series of simulated data** accuracy and loss vs epochs plotting: The figures at the top show maneuvers classification accuracy and loss plotting in PCNN, The figures at the bottom illustrate aggressive and normal behavior classification accuracy and loss plots in PCNN.

helps to improve accuracy in general. The comparison of accuracy results table illustrates that the LSTM method is not working as well as PCNN in terms of general accuracy, but when investigating from confusion matrix, for two classes (normal right turn and aggressive right lane-change), we find that LSTM works better than PCNN. Also, LSTM takes longer to train than PCNN. Similarly, while HMM does not work to detect aggressive maneuvers classes (general accuracy is approximately 20%), it gives the highest single accuracy for normal left and right turns classification among CNN, PCNN, and LSTM. However, we found that HMM results were very sensitive to any small changes in parameters. In general, CNN and PCNN are more efficient methods to detect essential features from aggressive behaviors comparing to the other methods considering both accuracy and cost. Moreover, a different feature engineering technique is applied on the simulated time series, that is angle of rotation to x and y locations. PCNN is applied on time series data which consists of these rotated locations, speed, and angle. This leads to significant improvement in results. Considering all 8 classes including both tasks, test accuracy jumps from 65.7% to 95% when position data rotated to a determined quadrant. The PCNN classification on different versions of dataset F-score results is shown in Table V. This f-score results are means of validated model by training PCNN model 10 times on simulated dataset. For each single training, randomly 20%

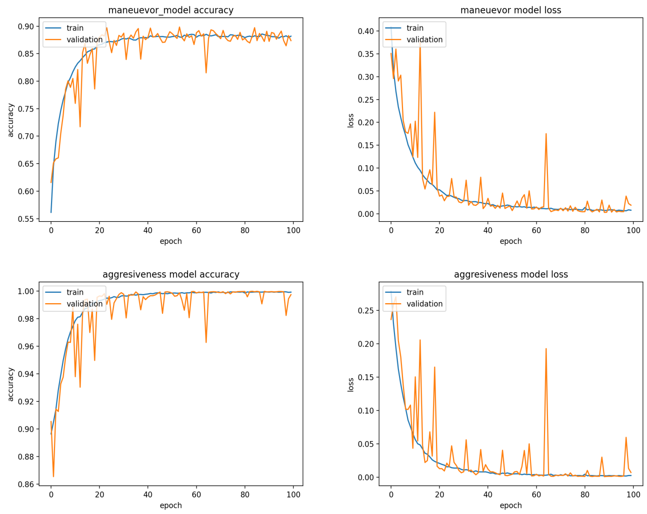

Fig. 10: PCNN classification on **real driven dataset** accuracy and loss vs epochs plotting: The figures at the top show maneuvers classification accuracy and loss plotting in PCNN, The figures at the bottom illustrate aggressive and normal behavior classification accuracy and loss plots in PCNN.

Method	Train accuracy(%)	Test accuracy(%)
CNN	69.5	60.6
PCNN	71.8	65.7
LSTM	65.8	58.13
HMM	55.28	51.22
SVM	63.4	60.5

TABLE IV: General accuracy results of classification for 8 classes together with various models applied on whole **plain simulated time series dataset**.

of selected for testing, from the training set 20% has been selected for validation. The early stopping point is determined as 11 epochs and 100 sample sequences has been used for an epoch. Datasets which are applied of different window sizes (300 and 200) has also been compared in this table. In order to prevent overfitting, the best epoch number for training PCNN can be selected between 10 and 15 based on accuracy and loss results which are shown in Figure 9.

After that, the PCNN model is performed on time series data from DBL. Location, speed and heading angle information are utilized as input. While CNN shows great performance in accuracy for single task, PCNN gives significant results in accuracy for multi tasks on DBL. These results indicates that CNN has ability to gather high level information from time series efficiently. PCNN multitask classification model's accuracy/loss vs epoch plot at Figure 10 illustrates that 99% accuracy in behavior classification, 86% accuracy in maneuver classification for 3 sec window size. Stopping epoch numbers can be selected from between 40 and 60 based on these plots as well. The PCNN classification on different versions of DBL dataset F-score results is shown in Table V. These f-score results are means of validated model by training 10 times PCNN model on the dataset.

Method	**Manoeuvre Classification** F1 score	**Behavior Classification** F1 score	**Manoeuvre Classification** F1 score	**Behavior Classification** F1 score
Time window size	3sec	3sec	2sec	2sec
PCNN Simulated Time Series Data	0.75	0.65	0.72	0.51
PCNN on Simulated Time Series Data with Rotated x and y	**0.97**	0.94	**0.96**	0.90
PCNN on DBL Time Series Data	0.87	**0.98**	0.85	**0.95**

TABLE V: PCNN application results comparison table on the different type of simulated data

VI. Conclusion

The contributions of this paper are: (1) Introduction of automated data production by the SUMO and Webots simulators for a set of random road configurations and various driver behaviors, (2) Real-world driving dataset application, (3) Applying CNNs on time series for driver behavior analysis, and (4) Introduction of a parallel CNN architecture with a shared early stage and split later stages for dual training of behavior and maneuver classifiers. We proposed Parallel Convolutional Neural Networks to classify aggressive driver behaviors together with maneuver classification. Two CNNs in the proposed method were trained and tested separately on large simulated and real-world driving datasets. These experiments show that these CNN models are able to capture high-level features to differentiate driving maneuvers and aggressive driver behaviors. Moreover, the results illustrate that CNN classifications can work in parallel and can support each others efficiently. This initial step was important for designing PCNN to build driving assist systems using embedded smartphone sensors. As a future work, the plan is to apply the designed PCNN to entire DBL dataset which includes 61 participants.

Acknowledgment

This project was possible through the financial support of grants from the National Science and Engineering Research Council (NSERC) of Canada (Grant #: CRDPJ 519912 – 17) and through financial support through our industry partner Magna International Inc.

References

[1] "Speeding." [Online]. Available: https://www.nhtsa.gov/risky-driving/speeding

[2] G. H. Walker and N. A. Stanton, *Human factors in automotive engineering and technology*. CRC Press, 2017.

[3] Q. Deng and D. Söffker, "A review of the current hmm-based approaches of driving behaviors recognition and prediction," *IEEE Transactions on Intelligent Vehicles*, 2021.

[4] R. Fuller, "Towards a general theory of driver behaviour," *Accident Analysis & Prevention*, vol. 37, no. 3, pp. 461–472, 2005.

[5] J. Pérez, D. Gonzalez, and V. Milanés, "Vehicle control in ADAS applications: State of the art," *Intelligent Transport Systems: Technologies and Applications*, p. 206, 2015.

[6] D. Chen, K.-T. Cho, S. Han, Z. Jin, and K. G. Shin, "Invisible sensing of vehicle steering with smartphones," in *Proceedings of the 13th Annual International Conference on Mobile Systems, Applications, and Services*. ACM, 2015, pp. 1–13.

[7] I. Mohamad, M. A. M. Ali, and M. Ismail, "Abnormal driving detection using real time global positioning system data," in *Space Science and Communication (IconSpace), 2011 IEEE International Conference on*. IEEE, 2011, pp. 1–6.

[8] J. F. Júnior, E. Carvalho, B. V. Ferreira, C. de Souza, Y. Suhara, A. Pentland, and G. Pessin, "Driver behavior profiling: An investigation with different smartphone sensors and machine learning," *PLoS one*, vol. 12, no. 4, p. e0174959, 2017.
[9] A. Ng, J. Ngiam, C. Y. Foo, Y. Mai, and C. Suen, "Unsupervised feature learning and deep learning," 2013.
[10] N. Lin, C. Zong, M. Tomizuka, P. Song, Z. Zhang, and G. Li, "An overview on study of identification of driver behavior characteristics for automotive control," *Mathematical Problems in Engineering*, vol. 2014, 2014.
[11] Z.-J. Cheng, L.-W. Jeng, and K. Li, "Behavioral classification of drivers for driving efficiency related ADAS using artificial neural network," in *2018 IEEE International Conference on Advanced Manufacturing (ICAM)*. IEEE, 2018, pp. 173–176.
[12] I. G. Daza, N. Hernández, L. M. Bergasa, I. Parra, J. J. Yebes, M. Gavilán, R. Quintero, D. F. Llorca, and M. Sotelo, "Drowsiness monitoring based on driver and driving data fusion," in *Intelligent Transportation Systems (ITSC), 2011 14th International IEEE Conference on*. IEEE, 2011, pp. 1199–1204.
[13] F. Tango and M. Botta, "Real-time detection system of driver distraction using machine learning," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 2, pp. 894–905, 2013.
[14] Y. Liu, X. Wang, L. Li, S. Cheng, and Z. Chen, "A novel lane change decision-making model of autonomous vehicle based on support vector machine," *IEEE access*, vol. 7, pp. 26 543–26 550, 2019.
[15] W. Wang, J. Xi, A. Chong, and L. Li, "Driving style classification using a semisupervised support vector machine," *IEEE Transactions on Human-Machine Systems*, vol. 47, no. 5, pp. 650–660, 2017.
[16] R. Kalsoom and Z. Halim, "Clustering the driving features based on data streams," in *Multi Topic Conference (INMIC), 2013 16th International*. IEEE, 2013, pp. 89–94.
[17] V. Gadepally, A. Krishnamurthy, and U. Ozguner, "A framework for estimating driver decisions near intersections," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 2, pp. 637–646, 2014.
[18] P. Boyraz, M. Acar, and D. Kerr, "Signal modelling and hidden markov models for driving manoeuvre recognition and driver fault diagnosis in an urban road scenario," in *Intelligent Vehicles Symposium, 2007 IEEE*. IEEE, 2007, pp. 987–992.
[19] O. Derbel *et al.*, "Driving style assessment based on the gps data and fuzzy inference systems," in *2015 IEEE 12th International Multi-Conference on Systems, Signals & Devices (SSD15)*. IEEE, 2015, pp. 1–8.
[20] J. Cervantes-Villanueva, D. Carrillo-Zapata, F. Terroso-Saenz, M. Valdes-Vela, and A. F. Skarmeta, "Vehicle maneuver detection with accelerometer-based classification," *Sensors*, vol. 16, no. 10, p. 1618, 2016.
[21] J. Ferreira, E. Carvalho, B. V. Ferreira, C. de Souza, Y. Suhara, A. Pentland, and G. Pessin, "Driver behavior profiling: An investigation with different smartphone sensors and machine learning," *PLoS one*, vol. 12, no. 4, p. e0174959, 2017.
[22] J. A. C. Dias, P. Machado, and F. C. Pereira, "Simulating the impact of drivers' personality on city transit."
[23] S. M. Hegazy and M. N. Moustafa, "Classifying aggressive drivers for better traffic signal control," in *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2017, pp. 702–707.
[24] P. A. Lopez, M. Behrisch, L. Bieker-Walz, J. Erdmann, Y.-P. Flötteröd, R. Hilbrich, L. Lücken, J. Rummel, P. Wagner, and E. Wießner, "Microscopic traffic simulation using sumo," in *The 21st IEEE International Conference on Intelligent Transportation Systems*. IEEE, 2018. [Online]. Available: https://elib.dlr.de/124092/
[25] W. Wang, W. Yang, A. Chen, and Z. Pan, "A scoring method for driving safety credit using trajectory data," *arXiv preprint arXiv:1811.12223*, 2018.
[26] D. Johnson, M. M. Trivedi *et al.*, "Driving style recognition using a smartphone as a sensor platform," in *Intelligent Transportation Systems (ITSC), 2011 14th International IEEE Conference on*. IEEE, 2011, pp. 1609–1615.
[27] X. Liu, H. Mei, H. Lu, H. Kuang, and X. Ma, "A vehicle steering recognition system based on low-cost smartphone sensors," *Sensors*, vol. 17, no. 3, p. 633, 2017.