Lane change behaviour inference through deep-learning-based environment analysis

Alberto Díaz-Álvarez[1*], Edgar Talavera-Muñoz[2], Felipe Jiménez Alonso[3], Francisco Serradilla García[1], Cristina Olaverri-Monreal[4], Álvaro San Juan Cervera[3]

1. Dpto. Inteligencia Artificial, ETSI de Sistemas Informáticos, Universidad Politécnica de Madrid, Spain; alberto.diaz@upm.es, francisco.serradilla@upm.es.

2. Dpto. Sistemas Informáticos, ETSI de Sistemas Informáticos, Universidad Politécnica de Madrid, Spain; e.talavera@upm.es.

3. University Institute for Automobile Research (INSIA), Universidad Politécnica de Madrid, Spain; felipe.jimenez@upm.es, alvaro.sancervera@alumnos.upm.es.

4. Chair for Sustainable Transport Logistics 4.0, Johannes Kepler University Linz, Austria; cristina.olaverri-monreal@jku.at.

Abstract

Sensor-equipped vehicles are the way to provide autonomy as they allow to perceive the environment. The usual sensors include, among others, LiDARs to perceive and give a sense of the surrounding environment to the algorithms implemented on it. This paper proposes a way of inferring in real time the lane change intention in urban scenarios by using Convolutional Neural Networks (CNNs) and the environment where the driver is immersed. After a preliminary study conducted with real data, it is observed that the trained models are capable of mimicking lane change behaviours, depending on the environment that surrounds them, similar to what a real driver would do. The approach can therefore be considered as a way forward when it comes to incorporating human behaviour into autonomous vehicles, thus facilitating the transition period to fully autonomous driving.

Keywords:

Deep Learning, Convolutional Neural Networks, Lane Change Behaviour

Introduction

In the next few years, driving, as we know it will undergo a dramatic transformation. We live in an era where companies assess fully autonomous vehicles in real environments, recording more than 800,000 kilometres without a single incident [1]. However, during the transition period until reaching a 100% penetration rate, a series of measures to minimize risks and accidents will need to be adopted in order to integrate autonomous fleets in a mixed traffic flow.

Artificial intelligence techniques based on supervised learning schemes have experienced a gigantic boom in the last decade, due to the raise of Deep-Learning methods. In the area of Intelligent

Transportation Systems (ITS), techniques based on Artificial Neural Networks (ANNs), such as obstacle warning systems or quasi-autonomous driving systems, are becoming more and more popular.

This study will determine whether and to what extent it is possible to implement lane change manoeuvres that accurately reproduce human driving patterns by using Convolution Networks, a type of ANN being used in Machine Learning that can extract spatial data patterns.

Lane change models

Although there were prior ones, the Gipps model [1, 2] was a breakthrough in lane change behaviour modelling. It proposed a behavioural car-following model describing the longitudinal interaction between vehicles that considered the integration of lateral models. This model was extended by many authors to study different approaches, such as differentiation between slow and fast lanes [3], communication models for collaborative lane changes [4] or probabilistic decision trees [5].

In 2000 Naturalistic Driving Data began to be used for the training of longitudinal models based on ANN [6, 7]. Other noteworthy works are [8], where Elfman Networks are compared with Multilayer Perceptrons for their ability to infer time patterns, [9] where Fuzzy Logic models are used, or [10] where behaviours are adjusted via genetic algorithms.

ANNs and Fuzzy Logic are not the only computational intelligence (CI) techniques used to model behaviours. Other works present models to characterize drivers based on Bayesian networks [11, 12], clustering techniques (through Latent Dirichlet Allocation) [13] or Hidden Markov Models [14, 15].

Convolutional Neural Networks

One of the most popular techniques in computational intelligence nowadays are the Convolutional Neural Networks (CNNs). Thanks to the rise of Deep Learning, they are one of the great exponents when it comes to classifying images, especially working with n-dimensional feature maps. As its very name suggests, they are ANNs that use a combination of two images to form a third one (convolutions) for their operation. Its operation is different to the Multilayer Perceptrons as its topology is organized into two well-differentiated regions, one that is dedicated to the extraction of input characteristics (*pattern extraction region*) and other to the output classification or regression given the extracted features (*inference region*).

CNNs are currently mainly used for trajectory tracking and prediction [16], lane change identification [17], or driver characterization. They are also used to model behaviour, specifically for lane change, both in decision making [18] and in their way of execution [19].

Problem formulation

The lane change problem determines *when* and *how* a driver performs a lane change manoeuvre at a given time. The intuition about the problem is that many of the key factors are not measurable in the real world (e.g. moods, physical condition, fortuitous events, etc.). In a previous paper [19] the authors tried to control these factors by dividing the problem into two parts, the *intention* and the *execution* of the lane-change, fixing the former and modelling the latter. This allowed studying how the various driver profiles executed lane changes in different ways and how this phenomenon can be modelled.

In this work we present the model of the whole lane change manoeuvre in an urban environment, i.e. deciding whether to change a lane by considering the driving environment through the measurable variables. We first approach the problem by using classification, to maximize the number of matches between the real lane changes and those predicted by the model.

Models will be trained with a set of driving data acquired in a real driving environment. The possibility of making predictions about future lane change manoeuvres by analysing available driving patterns from real data will then be examined by determining the best number of independent variables or potential predictors for our dependent variable lane change manoeuvre. and finally, the best matching CNN architecture to the results obtained will be determined. This process will provide a a foundation of the sensory input that intelligent agents (IA) in autonomous vehicles require to perform the correspondent driving action that relates to lane change.

Methodology

We propose to capture driver data from both the environment and the state of the vehicle on two different urban routes. These routes, from now on R_1 and R_2, will be considered equivalent as they are roads in an urban environment, with sections of between one and three lanes along the route and with maximum speeds established between $30kmh^{-1}$ and $50kmh^{-1}$. R_1 has an estimated travel time of $30min$ and will be used as a data source for model training (training and validation datasets). R_2 has an estimated travel time of $15min$ and its data is intended to serve as a test set. Both routes have been carried by three different subjects (male, 30 to 35 years old and more than six years of driving experience). The driving tests were made between 11:00 am and 12:00 pm on weekdays, allowing a road traffic with enough vehicles to require lane change manoeuvres.

An instrumented vehicle, more specifically a Mitsubishi iMiEV, has been used with the following devices attached: a **CAN Reader**, to retrieve the internal state of the vehicle, a **camera** to provide a frontal view of the vehicle, a **GPS** to capture standard NMEA GGA (geopositioning) and VTG (speeds) messages and a 16-channel **LiDAR**.

All devices are connected to a computer with Debian 9.5 GNU/Linux operating system on an Intel i7-7500U CPU with 16GB of RAM. The data capture software has been developed on ROS, and each device has its own capture node to transfer the information to the general data repository (Figure 2).

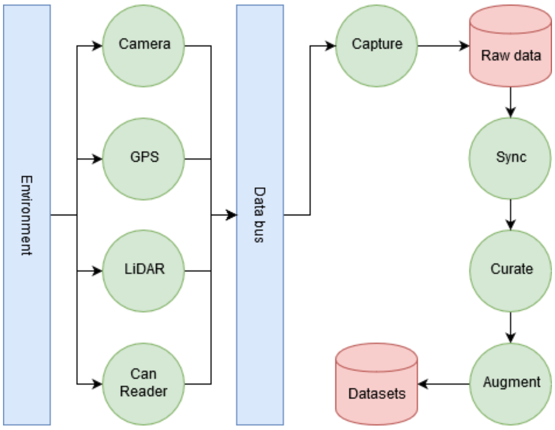


Figure 2 – Data schematics for the instrumented vehicle and the data post-processing pipeline.

Data

Each sensor sends its own messages at a different rate, so the first step was fusing the different information at the same rate, in our case $10\ Hz$. All the collected data and its sources are depicted in Figure 3, also indicating whether a manual post-processing was required after the synchronization step.

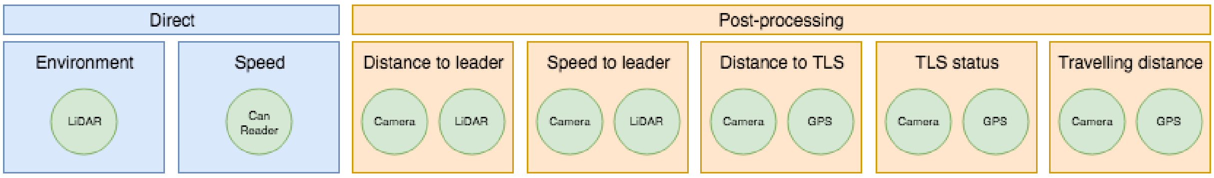


Figure 3 – Data schematics for the instrumented vehicle and the data post-processing pipeline.

The required description of the environment is deduced from the point cloud that is extracted from the LiDAR. These data required an additional post-processing as explained below

1) CNNs requires a fixed sized input, but the point cloud has a variable number of points. Thus, depth maps (Figure 4) are required to represent the environment as an image of a single channel where each pixel is the distance to a spherical sector of the original space. The points will therefore be generated with a 1^o horizontal resolution, ranging the six channels from -7^o to 3^o (a smaller angle denotes the impact with the roof of the car, and a larger one denotes non-relevant information on the surrounding traffic). The process generates a 6×360 depth map.

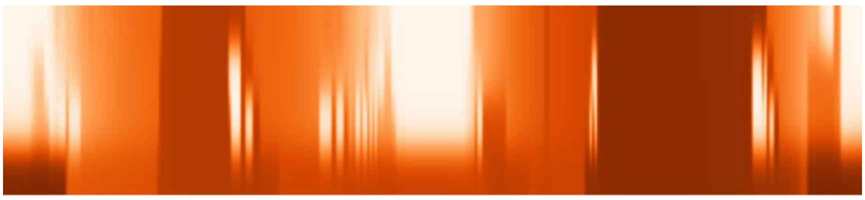
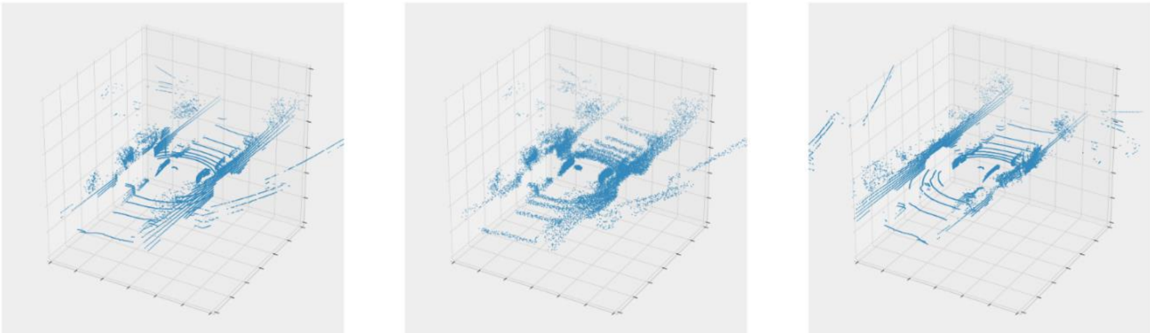


Figure 4 - Depth map as a grayscale image (orange tinted and blurred to improve figure appreciation).

2) The point cloud is produced by a single laser attached to a mechanical device that operates with

spherical coordinates at a horizontal resolution of 0.2^o and vertical resolution of 2^o, so the unobservable surface is larger as we move away from the origin. We have then considered $25m$ as the limit beyond which the elements cease to be recognisable. These will be the values of the depth map matrix, normalized to the interval $[0,1]$.

The data transformation pipeline includes a data augmentation process. This is because the rate of lane changes is very low compared to the rest of the actions, in which there are not lane changes. Therefore, there is an evident bias of data towards no lane change. wo data augmentation techniques (*mirroring* and *shaking,* Figure 5) are used in this work to increase lane change actions and reduce the existing bias. *mirroring* generates a new point cloud for each point cloud, through symmetry with respect to the XY plane. The other approach, *shaking*, generates a new point cloud. It with noise.



(a) (b) (c)

Figure 5 – A representation of (a) a point cloud, (b) the same point cloud after a shake of 1cm, and (c) the mirror image of the original.

As the process of applying convolutional neural nets to time series implies missing an intuitional sense for time series data, we overcome this limitation, relying on the suggestion in [19] and selected as input for the models temporary frames, specifically the times t_0 (current moment) t_{10} (previous moment to the current one, 1 second before) and t_{20} (previous moment to the current one, 2 seconds before), Including three temporal moments helps models to know intuitively the patterns that correspond to the first and second moment derived from the position (speed and acceleration).

In order to be able to fusion spatial data (depth map) with non-spatial data (additional input data) we slightly modified the CNN architecture. The depth maps were processed by the feature extraction region, and the rest of the parameters were inserted directly into the inference region.

In addition, to minimize the loss of spatial data, the sides of the depth maps were enlarged with the opposite ends of the map itself, with a size of a half of the convolution filter applied in the first layer. In this way, the ends of the image are always analysed, instead of losing that region (Figure 6).

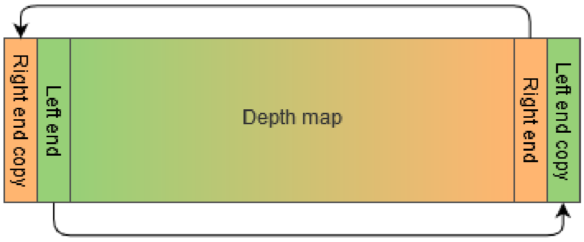


Figure 6 - The extensions prevent the loss of information at the ends of the image (the rear of the vehicle).

Methodology

After the data processing, two datasets are available (see Table 1), one corresponding to route R_1 (training and validation sets) and R_2 (test set),

Table 1 – Dataset description.

Dataset	Size	Lane changes		
		Left	None	Right
Training	248930	12740	0,576	12740
Test	82060	1211	0,569	533

To minimize training and validation errors a series of training processes are applied on different CNN architectures. After the process, the best network will be further analysed to verify how much it differs from a human behaviour model.

The training was performed on an Intel®Core i7-6700K computer at 4.00 GHz and 16 GiB of memory, with a Titan X GPU granted by NVIDIA. The operating system was a Debian GNU/Linux version 9.6. The training scheme between models coincides. Because the size of the training and validation sets was large, they were divided into subsets. Each training batch contained then a random selection of the sample equally distributed among the three classes of the solution in order to avoid bias. The training algorithm used was ADAM, with the cross entropy as cost function, and with ReLU type neurons except in the last layer, which maintained a linear activation scheme followed by a Softmax normalization layer (as the label classes are mutually exclusive).

To reduce overfitting and to improve the generalization, randomly selected neurons were ignored during training, following a dropout approach with a probability of 0.1. The training process was stopped after 10^6 epochs.

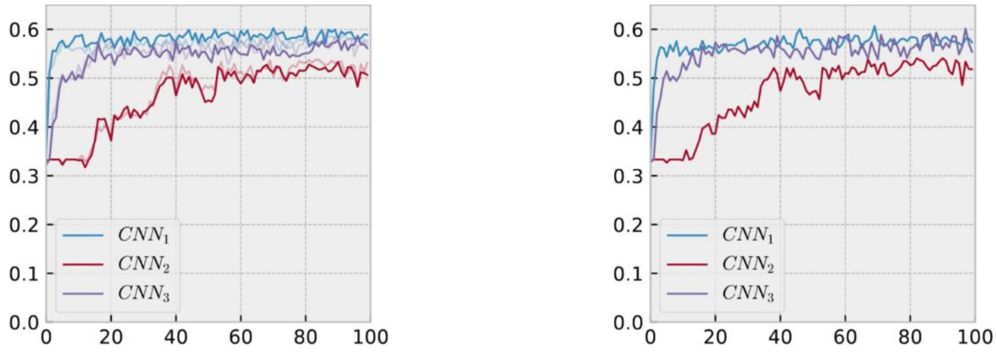
Results

After a training process of 22 architectures, the best results after the tests are described in Table 2.

Table 2 – Top-3 CNN architectures.

Network	Topology
CNN_1	c16-4-18-v d128
CNN_2	c64-5-36-v c256-3-5-v d256-d128-d16
CNN_3	c16-3-18-v c32-3-18-v c64-2-18-v d128

Being $cF\text{-}W\text{-}H$ a convolution layer with F size filters of $W \times H$ and dN a fully connected layer of N neurons. Figure 7 shows the evolution of the accuracy during training.



(a) (b)

Figure 7 - Evolution of accuracy by time for (a) training, validation and (b) test sets. X-axis indicate the time (in thousands of epochs) and the ordered the precision reached.

A relatively large network was needed to exceed the limit imposed by the random classification. From these the CNN_1 model is the one that gives the best results in the training phase. The horizontal lines show the error in the test set of the architectures. The specific values are shown in Table 3.

Table 3 - Accuracy of trained models with training, validation and test sets.

Network	Accuracy		
	Training	Validation	Test
CNN_1	0.589	0.576	0.573
CNN_2	0.506	0.531	0.518
CNN_3	0.561	0.569	0.554

Confusion matrices were to check the specific types of errors associated with misclassifications. Figure 8 shows the matrices for immediate prediction and for $2.5s$ ahead prediction, time that we considered enough to associate a lane change with an action, pattern, behaviour? that was known intuitively by the network.

Real \ Predicted	L	N	R
L	32.44	47.11	20.45
N	5,76	60.50	33.74
R	12.31	70.30	17.39

(a)

Real \ Predicted	L	N	R
L	42.86	36.69	20.45
N	5.76	60.50	33.74
R	12.31	45.13	42.56

(b)

Figure 8 - Confusion matrices for (a) instant prediction, and (b) after 2.5 seconds (units in %).

As previously mentioned, the results indicated a slight bias of the network towards a no lane change behaviour when it came to immediate prediction. However, increasing the time window to $2.5s$, also increased the accuracy on the lane changes prediction This calculation was been made by counting as matches only those cases where there was only one change to the desired lane, and not those cases in which the change is subsequent to a change to the opposite lane.

Conclusions

We can conclude that the obtained results in this work are promising. On one hand, the models mimic human behaviour in similar surrounding environments. Therefore, they can be used to reproduce driving behavioural patterns of drivers in a real world setting with autonomous vehicles. Future work will aim at enhancing the present approach by increasing the number of subjects who participated in the study. Further, a reduction of noise will be striven through an increment in the amount of artificial data generated. According to [19], a substantial improvement in quality was observed by increasing the data with the two techniques described. In future research, the data set will increasingly incorporate routes to calibrate the extent to which these augmentation techniques are no longer having a positive impact. Another interesting aspect for the future is the evaluation in extra-urban environments, where speeds are higher, manoeuvres are different, and lane changes to the left or right differ.

Acknowledgements

This work has been partially funded by the spanish Ministerio de Economía y Competitividad MINECO (CAV Project TRA2016-78886-C3-3-R) financed this work. We would also like to thank NVIDIA for generously providing us with the Titan X GPU used to perform this study.

References

1. Gipps, P. G. (1981). A behavioural car-following model for computer simulation. *Transportation Research Part B: Methodological*, *15*(2), 105-111.

2. Gipps, P. G. (1986). A model for the structure of lane-changing decisions. *Transportation Research Part B: Methodological*, *20*(5), 403-414.
3. Weidemann, R., & Reiter, U. (1992). Microscopic Traffic Simulation, The Simulation System-Mission. *University Karlsruhe, Germany*.
4. Hidas, P. (2002). Modelling lane changing and merging in microscopic traffic simulation. *Transportation Research Part C: Emerging Technologies*, *10*(5-6), 351-371.
5. Toledo, T., Koutsopoulos, H. N., & Ben-Akiva, M. (2007). Integrated driving behavior modeling. *Transportation Research Part C: Emerging Technologies*, *15*(2), 96-112.
6. Hongfei, J., Zhicai, J., & Anning, N. (2003, October). Develop a car-following model using data collected by" five-wheel system". In *Intelligent Transportation Systems, 2003. Proceedings. 2003 IEEE* (Vol. 1, pp. 346-351). IEEE.
7. Panwai, S., & Dia, H. (2007). Neural agent car-following models. *IEEE Transactions on Intelligent Transportation Systems*, *8*(1), 60-70.
8. Simonelli, F., Bifulco, G. N., De Martinis, V., & Punzo, V. (2009). Human-like adaptive cruise control systems through a learning machine approach. In *Applications of Soft Computing* (pp. 240-249). Springer, Berlin, Heidelberg.
9. Wu, J., Brackstone, M., & McDonald, M. (2003). The validation of a microscopic simulation model: a methodological case study. *Transportation Research Part C: Emerging Technologies*, *11*(6), 463-479.
10. Naranjo, J. E., Jiménez, F., Serradilla, F. J., & Zato, J. G. (2012). Floating car data augmentation based on infrastructure sensors and neural networks. *IEEE Transactions on Intelligent Transportation Systems*, *13*(1), 107-114.
11. Maye, J., Triebel, R., Spinello, L., & Siegwart, R. Y. (2011). Bayesian on-line learning of driving behaviors. In *2011 IEEE International Conference on Robotics and Automation,(ICRA 2011)* (pp. 4341-4346). IEEE.
12. Van Ly, M., Martin, S., & Trivedi, M. M. (2013, June). Driver classification and driving style recognition using inertial sensors. In *Intelligent Vehicles Symposium (IV), 2013 IEEE* (pp. 1040-1045). IEEE.
13. Bender, A., Agamennoni, G., Ward, J. R., Worrall, S., & Nebot, E. M. (2015). An unsupervised approach for inferring driver behavior from naturalistic driving data. *IEEE Transactions on Intelligent Transportation Systems*, *16*(6), 3325-3336.
14. Kuge, N., Yamamura, T., Shimoyama, O., & Liu, A. (2000). *A driver behavior recognition method based on a driver model framework* (No. 2000-01-0349). SAE Technical Paper.
15. Sekizawa, S., Inagaki, S., Suzuki, T., Hayakawa, S., Tsuchida, N., Tsuda, T., & Fujinami, H. (2007). Modeling and recognition of driving behavior based on stochastic switched ARX model. *IEEE Transactions on Intelligent Transportation Systems*, *8*(4), 593-606.
16. Deo, N., & Trivedi, M. M. (2018). Convolutional Social Pooling for Vehicle Trajectory Prediction. *arXiv preprint arXiv:1805.06771*.

17. Ye, Y. Y., Hao, X. L., & Chen, H. J. (2018). Lane detection method based on lane structural analysis and CNNs. *IET Intelligent Transport Systems*.
18. Hoel, C. J., Wolff, K., & Laine, L. (2018). Automated Speed and Lane Change Decision Making using Deep Reinforcement Learning. *arXiv preprint arXiv:1803.10056*.
19. Díaz-Álvarez, A., Clavijo, M., Jiménez, F., Talavera, E., & Serradilla, F. (2018). Modelling the human lane-change execution behaviour through Multilayer Perceptrons and Convolutional Neural Networks. *Transportation Research Part F: Traffic Psychology and Behaviour*, *56*, 134-148.