Imitation Learning for Autonomous Vehicle Driving: How Does the Representation Matter?

Antonio Greco(✉), Leonardo Rundo, Alessia Saggese, Mario Vento, and Antonio Vicinanza

Department of Information Engineering, Electrical Engineering and Applied Mathematics, University of Salerno, 84133 Fisciano, SA, Italy
{agreco,lrundo,asaggese,mvento,anvicinanza}@unisa.it

Abstract. Autonomous vehicle driving is gaining ground, by receiving increasing attention from the academic and industrial communities. Despite this considerable effort, there is a lack of a systematic and fair analysis of the input representations by means of a careful experimental evaluation on the same framework. To this aim, this work proposes the first comprehensive, comparative analysis of the most common inputs that can be processed by a conditional imitation learning (CIL) approach. With more details, we considered the combinations of raw and processed data—namely RGB images, depth (D) images and semantic segmentation (S)—to be assessed as inputs of the well-established Conditional Imitation Learning with ResNet and Speed prediction (CILRS) architecture. We performed a benchmark analysis, endorsed by statistical tests, on the CARLA simulator to compare the considered configurations. The achieved results showed that RGB outperformed the other monomodal inputs, in terms of success rate on the most popular benchmark NoCrash. However, RGB did not generalize well when tested on different weather conditions; overall, the best multimodal configuration was a combination of the RGB image and semantic segmentation inputs (i.e., RGBS) compared to the others, especially in regular and dense traffic scenarios. This confirms that an appropriate fusion of multimodal sensors is an effective approach in autonomous vehicle driving.

Keywords: Autonomous vehicle driving · Imitation learning · Conditional imitation learning · Benchmarking · CARLA

1 Introduction

Driving statistics confirm that distracted behavior is one of the main causes of the car accidents. Despite the effort of an experienced driver, keeping thorough attention on the road may be difficult, in particular during long travels [15]. This has motivated in the last decades a growing interest towards the design and development of autonomous vehicles. Anyway, nowadays this seems to be

S. Sclaroff et al. (Eds.): ICIAP 2022, LNCS 13231, pp. 15–26, 2022.
https://doi.org/10.1007/978-3-031-06427-2_2

feasible just in a limited set of scenarios, since the presence of dynamic obstacles in an unconstrained environment and the need to control the vehicle in real-time make this task particularly challenging. As pointed out in [20], the main techniques currently available for designing autonomous vehicles can be divided into modular and end-to-end (E2E) approaches. In a modular approach, the whole problem is partitioned into single tasks (e.g., data analysis, local planning, behavioral planning, motion planning). On the one hand, the advantage is the complete knowledge and control of each sub-task, thus allowing us to diagnose and solve possible driving errors. On the other hand, it requires a huge effort in the design of a system that takes into account any possible scenario. The E2E approaches drastically reduce the design effort by treating the whole driving problem as a single learning task despite a relevant amount of data would be necessary. In this field, we are experiencing a particular interest towards the imitation learning (IL) techniques, which allow us to train a policy learning from driving examples provided by an expert. For example, during the training phase, the model can be fed with raw data directly collected from sensors—such as an RGB image—or processed data—such as a semantic segmentation of the environment—and then it yields either the controlling value of the system—such as throttling, brake and steer values—or a set of waypoints that describe the local trajectory. The driving examples of the expert, also known as demonstrations, are a collection of pairs of inputs and outputs of the policy. The simplicity of data annotation and the possibility of training policies on recorded data are the most important advantages of IL [1,5]. Therefore, in this paper we focus on the approaches based on IL, which are attracting the interest of the scientific community in the latest years [14].

Relying upon the analysis of the literature, we can observe that most of the papers introduced new architectures or different inputs, but there is a lack of a systematic and fair analysis of the input representations by performing the experimental evaluation on the same framework. In addition, we note that the CIL technique is the most promising and investigated E2E approach in recent years. According to these observations, in this paper we aim to contribute to the state-of-the-art on multi-input CIL by proposing:

1. a comprehensive, comparative analysis of the most common inputs and their combinations that can be processed by a CIL approach [6]. To this end, we analyzed raw and processed data, namely RGB images, depth images and semantic segmentation;
2. a benchmark analysis, endorsed by appropriate statistical tests, on the CARLA simulator [9], to compare the considered configurations. We executed the analysis of 6 input configurations over 450 benchmark episodes for a total time of 135 h of testing time.

This work is organized as follows: Sect. 2 summarizes the literature background. Section 3 describes the dataset and benchmarks, as well as provides a complete description of the architectures and configurations used in our work. Section 4 presents and extensively discusses the results obtained. Finally, in Sect. 5 we draw some conclusions and future work.

2 Related Work

The most relevant literature approaches are described in this section.

One of the first examples was proposed by Bojarski *et al.* [2], who introduced the use of convolutional neural networks (CNNs) for imitation learning applied to autonomous vehicle driving. This method can only perform simple tasks, such as lane following, because it has not any command (from user or from automatic navigation systems) as input that can make a decision at an intersection, and so it cannot perform a navigation task. In [3], the authors showed the regions of an input image that contribute to the prediction of new actions. This approach attempts to provide an explanation to the output of the self-driving policy. This was the first approach to the explainability problem in this field. The recent work in [7] used attention mechanisms to achieve the same purpose. The authors highlighted that their method achieved results comparable with the state-of-art and, at the same time, the output is explainable.

A temporal sequence of data as input of a driving policy was introduced by Xu *et al.* [18] in which a recurrent neural network (RNN) was used to predict a moving path. The temporal information took into account the whole sequence of the performed actions but the RNN made the model computationally heavier. A step forward for the E2E architectures was proposed by Codevilla *et al.* [5] with the introduction of the conditional imitation learning (CIL) technique. This framework aims to solve the ambiguity at the intersections that the previous approaches suffer. It introduced lateral and longitudinal controls that perform the navigation task and defined the 'high-level command' input that enables the interaction with the navigation system. A modified version of this method was presented in [6]. The authors pointed out that the approach proposed in [5] exhibited limitations due to the bias of the adopted driving dataset. They proposed to add an input branch, as well as a deeper convolutional backend network, to use the measured speed as additional feature. This approach can reduce the inertia problem, i.e., the difficulty of restarting the vehicle after a stop for any reason. The policy does not directly model the causal signals, so it confuses the causes that lead to the stop command. Further works based on CIL methods exploit the effectiveness of the architecture using different learning methodologies. Chen *et al.* [4] presented, to the best of our knowledge, the state-of-art approach on simulator benchmarks, such as NoCrash [6] and CARLA [9]. The method was based on an agent trained with privileged information obtained from a simulator and then adopted as an expert for training a second agent without additional information. It outperformed the previous model but it is difficult to train in a real environment due to the presence of the privileged agent that should be trained using additional information that is not directly available in a real-world environment because it requires a huge effort for the labeling. The method proposed by Ohn *et al.* [13] was based on a multi-expert policy. It was trained using multiple driving policies with an IL strategy and the combination of the policies was then optimized *via* a task-driven refinement. From the one hand, it achieved the performance comparable to [4] but without requiring an additional image labeling. On the other hand, it needed a simulator to execute the task-driven refinement with an on-policy training.

The previous approaches assessed the performance of an end-to-end approach by using data acquired from a single sensor. Nevertheless, an autonomous vehicle is equipped with several sensors and all of them can contribute to the navigation task. The use of raw or processed data may be an additional design choice. An in-depth analysis on the representation of the input data is required to fully exploit the huge number of information extracted by the various sensors. In the recent literature, the exploitation of limits about the CIL technique by varying the input data has attracted a huge interest. In [1], a semantic segmentation (S) image was used as input of the network and the authors showed the effectiveness of this representation by varying the number of output classes and the resolution of the representation but they limited this analysis only to this type of input. The authors of [10] used a multimodal approach based on camera and Lidar. They proposed an algorithm to extract a Polar Grid View representation from the Lidar and then used a mid-fusion scheme.

In [17], an analysis of a CIL method with the use of an RGB camera with depth information was reported. The work focused on the different sensor-fusion methods between an RGB image and a depth image and demonstrated the effectiveness of an early-fusion scheme that used the four channel image composed of RGB image and depth image (RGBD) as input of the neural network. However, the best performing method showed the following constraint: the size of the RGB and depth images have to be the same to align them. The use of an RGBD image was typical also in [12], where the authors introduced the semantic segmentation of the scene as an additional task. They obtained an improvement of the performance compared to the early-fusion RGBD of [17] but their approach required additional effort for providing the semantic segmentation of input images.

From an overview of the previous methods, we notice that the integration of multiple inputs (such as the depth image or semantic segmentation) into the CIL method generally produces an increase of the driving ability, thus obtaining the best performance in the navigation without the dynamic obstacle task of the CARLA Benchmark. A further improvement of performance is obtained with the use of a deeper backend architecture and providing an additional task as the estimation of the semantic segmentation.

3 Materials and Methods

3.1 Dataset

The dataset CARLA100 [6] contains about 100 h of driving collected on Town01 of the CARLA simulator. From a first analysis of the dataset, we noticed that only 25 h provide RGB, semantic segmentation and depth images and, therefore, we selected only this subset of data for our experiments. We used also additional information, such as the speed measurement and the high-level command provided by the user. This last information expresses the intention of a user to take a specific direction at an intersection, thus it implicitly describes different behaviors.

The high-level commands are listed in what follows:

- `Follow Lane`: the user is sufficiently far from an intersection and it continues to follow the lane;
- `Left`: the user expressed the intention of turning to left at the next intersection;
- `Right`: the user expressed the intention of turning to right at the next intersection;
- `Go Straight`: the user expresses the intention of going straight at the next intersection.

We analyzed the distributions of high-level commands over the episodes of the dataset obtaining Fig. 1a, where we can observe that the data are balanced.

A weather type is defined for each episode. In the whole dataset, there are the following weather conditions:

- Clear Noon;
- After Rain Noon;
- Heavy Rain Noon;
- Clear Sunset.

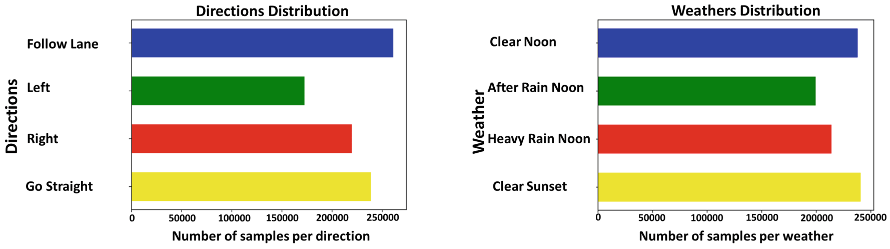
We analyzed also the distribution of the weather conditions over the episodes of the dataset in Fig. 1b, which shows a quite even distribution among the weather conditions.

We selected 20 h from the whole dataset preserving the *a priori* distribution of the high-level commands that allows for discriminating overall different scenarios (turns, lane following). We divided the dataset into 15 h for the training set and the remaining 5 h for the validation set.

3.2 Benchmark

The used benchmark was NoCrash [6], that is composed of three distinct tasks: Empty, Regular Traffic, and Dense Traffic. The tasks aim at emulating different traffic levels from an empty to a densely-populated town.

For each task, six different weather conditions were defined: the same of the training set and two additional ones (namely rainy after rain, soft rain sunset). For each pair $\langle \mathsf{Task}, \mathsf{Weather} \rangle$, the benchmark contains 25 episodes. Each of them consists of a path that the vehicle should travel across a specific CARLA town. An episode is considered successful when the vehicle completes the path without collisions and in a given time. The final measure is the success rate (expressed as a percentage) of episodes for each task provided by the driving benchmark. For result comparability with the literature, we performed the evaluation in Town01 and Town02.



(a) (b)

Fig. 1. Data distribution of the high-level commands (i.e., `Follow Lane`, `Left`, `Right`, `Go Straight`) and the weather conditions (i.e., Clear Noon, After Rain Noon, Heavy Rain Noon, Clear Sunset) over the samples of the dataset. **(a)** Data distribution of the high-level commands in the dataset; **(b)** Data distribution of the weather conditions in the dataset.

3.3 The Investigated CIL Method

The objective of imitation learning in the field of autonomous vehicle driving is the determination of a policy:

$$\pi : \mathcal{X} \rightarrow \mathcal{A}, \tag{1}$$

which links the input space $\mathcal{X}$, that can be composed of raw data collected from sensors, processed data or driving intentions, to a controlling space $\mathcal{A}$, defined in terms of driving commands or waypoints.

The policy is trained by minimizing a loss function $\mathcal{L}(\cdot,\cdot)$ between the model predictions and the demonstrations of an expert on the same input data. In this work, we adopt as output a three-dimensional controlling space composed of steering, throttling and braking commands, $\mathbf{a} = \langle s, a, b \rangle$. We also consider as additional task the prediction of the actual speed. The input space $\mathcal{X}$ depends on the specific configuration that we trained.

We considered the Conditional Imitation Learning with ResNet and Speed prediction (CILRS) architecture, introduced in [6], as our baseline, graphically represented in Fig. 2. It considers the predicted action $\mathbf{a}$ and the expert action $\mathbf{a}_{\text{expert}}$, and relies upon the loss function in Eq. (2), calculated for each sample:

$$\begin{aligned}\mathcal{L}(\mathbf{a}, \mathbf{a}_{\text{expert}}) &= \mathcal{L}\left(\langle v, s, a, b\rangle, \langle v_{\text{expert}}, s_{\text{expert}}, a_{\text{expert}}, b_{\text{expert}}\rangle\right)\\ &= w_1 \cdot ||v - v_{\text{expert}}|| + w_2 \cdot (\lambda_1 ||s - s_{\text{expert}}|| + \lambda_2 ||a - a_{\text{expert}}|| + \lambda_3 ||b - b_{\text{expert}}||),\end{aligned} \tag{2}$$

where w_1 and w_2 denote the weighting factors for the predicted speed and actual commands, respectively. Regarding the actual commands, $\lambda_1, \lambda_2, \lambda_3$ represent the weights for the terms s, a, b, respectively.

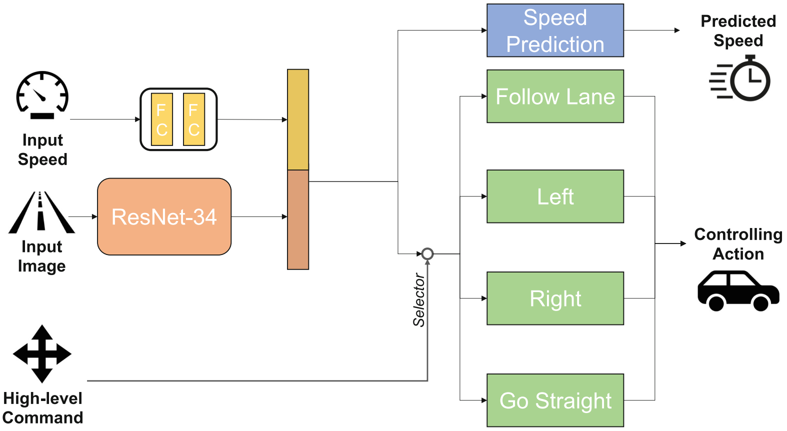


Fig. 2. General scheme of the CILRS architecture [6], which is used as our baseline for imitation learning. The investigated configurations can include as monomodal inputs: depth image, the estimation of a semantic segmentation, RGB image, or their multimodal combinations. The network takes two additional inputs: (*i*) the current speed, and (*ii*) the high-level command that acts as a switch and can select the output values to use. The branches are: `Follow Lane`, `Left`, `Right` and `Go Straight`. This mechanism is adopted to predict the controlling action aimed at performing different behaviors at intersections.

Considering the per-sample loss function $\mathcal{L}(\cdot,\cdot)$ in Eq. (2) and assuming the dataset as a set composed of N $\langle\mathsf{Observation},\mathsf{Action}\rangle$ pairs $\mathcal{D}=\{\langle\mathbf{o}_i,\mathbf{a}_i\rangle\}_{i=1}^{N}$, the objective of the imitation learning process is:

$$\underset{\theta}{\text{minimize}} = \sum_{i=0}^{N} \mathcal{L}(\tilde{\pi}(\mathbf{o}_i;\theta), \pi_{\text{expert}}(\mathbf{o}_i)), \tag{3}$$

where θ is the set of parameters of a function approximator $\tilde{\pi}(\mathbf{o}_i;\theta)$ of the expert's policy $\pi_{\text{expert}}(\mathbf{o}_i)$ for the i-th sample.

3.4 Analysis of the Input Representation

In particular, we propose to investigate three monomodal and three multimodal images to represent the input in an autonomous vehicle:

1. Depth (D);
2. Semantic Segmentation (S);
3. RGB image (RGB);
4. RGB and Semantic Segmentation (RGBS);
5. RGB and Depth (RGBD);
6. RGB, Depth and Semantic Segmentation (RGBDS).

All the previous configurations were fed to the same architecture illustrated in Fig. 2. In case of the three monomodal configurations D, S and RGB—which provide a single image as input data—we did not modify our baseline architecture. For the multimodal configurations—namely RGBS, RGBD and RGBDS—we adopted an early-fusion approach that demonstrated to be the best fusion method according to [17]. In this case, the baseline was modified by introducing an extra convolutional layer with kernel size of 1×1 to keep as output the same input dimension and reduce the feature layers.

For assessing statistical differences between the configurations, we used a Wilcoxon test [16] on paired results (i.e., the distribution metric values for all the samples of the dataset). In all the tests, a significance level of 0.05 was considered. The computed p-values were then corrected using the Bonferroni-Holm method for multiple comparisons [11].

3.5 Training Procedure

The models were trained using the Adam optimizer with a batch size of 128 and an initial learning rate of 0.0002. We performed a min-max scaling, in the $[0, 1]$ range, of the input images. With the RGB image inputs, the following data augmentation transformations were applied: Random Hue Variation, Add Shadow, Add Fog, Darken, Brighten. The initial weights were pretrained on ImageNet [8]. In the loss function, the weighting factors were set as follows: $w_1 = 0.08, w_2 = 0.92$ and $\lambda_1 = 0.50, \lambda_2 = 0.45, \lambda_3 = 0.05$ according to [6].

All the models were implemented in Python using TensorFlow version 1.14.

4 Experimental Results

The methods were evaluated on the NoCrash [6] benchmark on Town01 and Town02. Table 1 shows the achieved results in terms of success rate.

The evaluation of the different configurations on the benchmark shows that among the monomodal inputs, D and S are not able to drive in a dense and new scenario. However, the Depth configuration did not generalize well and obtained the worst performance over all the configurations in presence of dynamic obstacles. This might be due to the lack of information on the surrounding environment, thus demonstrating difficulties to distinguish between the obstacles and the road. The semantic segmentation alone achieved low performance for all the benchmark tasks. The monomodal RGB configuration showed quite good driving abilities in an already known scenario but it did not generalize well in presence of different weather conditions. Although the adopted data augmentation should reduce this effect, over-specialization might still remain.

The multimodal inputs obtained overall the best results. Actually, they are able to effectively exploit the information conveyed by each sensor. From our analysis, the best combination of inputs is RGBS. It achieves the best performance on six tasks over the different scenarios against the four best values

Table 1. Results achieved by the investigated configurations. Each row contains the success rate (in percentage) of the episodes for each task in a specific scenario. The Training scenarios contain the results obtained on Town01 with the training weathers. The New Weather scenarios were obtained on Town01 with the testing weathers. The New Town scenarios were the results obtained on Town02 with the training weathers, while the New Town & Weather scenarios contain the results obtained on Town02 with the testing weathers. The highest performance for each setting is highlighted in bold.

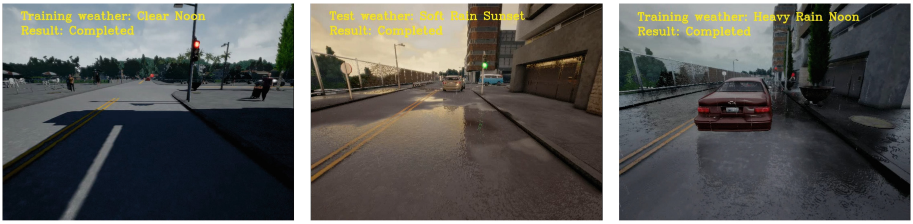
Scenario	Task	D	S	RGB	RGBS	RGBD	RGBDS
Training	Empty	83	13	80	**97**	96	**97**
	Regular	59	21	70	**94**	83	86
	Dense	9	23	26	**58**	43	47
New Weather	Empty	80	10	92	96	92	**98**
	Regular	60	14	76	**94**	78	**94**
	Dense	10	26	44	**52**	40	38
New Town	Empty	11	8	27	**95**	69	90
	Regular	5	20	16	**83**	48	77
	Dense	0	8	6	29	16	**30**
New Town & Weather	Empty	10	10	8	74	48	**86**
	Regular	4	18	4	70	48	**80**
	Dense	0	6	2	**28**	10	26
Average	Empty	46.00	10.25	51.75	90.50	76.25	**92.75**
	Regular	32.00	18.25	41.50	**85.25**	64.25	84.25
	Dense	4.75	15.75	19.50	**41.75**	27.25	35.25

obtained by RGBDS. The evaluation of the average performance for each driving task confirms the previous trend. The RGBD configuration overcomes the monomodal inputs, but it is beyond the other multimodal combinations. Furthermore, we consider that the depth image can be obtained directly from a depth sensor, instead the semantic segmentation requires an additional computational effort for its estimation. Considering the trade-off between performance and prediction time, a well-established network is the Bilateral Segmentation Network (BiSeNet) V2 [19] takes on average 0.083 s on a NVIDIA Jetson Xavier AGX with a mean Intersection over Union (mIoU) of 72.6% for a prediction.

According to Table 1, all pairwise comparisons, based on the Wilcoxon tests, showed statistically significant differences ($p \ll 0.001$ after the Bonferroni-Holm correction).

We recorded the episodes performed by our best model on the CARLA Simulator, as shown in Fig. 3. From the recorded video, we identified two problems: the former is the jerky driving (i.e., non-smooth accelerations); the latter is related to the well-known inertia problem (i.e., when the vehicle stops or tends to remain stopped).

We tested the best configuration obtained on a real-size autonomous vehicle. We performed a fine-tuning on our real dataset acquired in our University campus and then we selected a set of well-known paths. From our experience, the autonomous vehicle is able to effectively perform the lane following task. Moreover, it stops in presence of pedestrians or other vehicles and it also reduces its speed at the crosswalks. However, we identified the same problems of the simulation environment: when its speed is 0, it is slow to restart, thus exhibiting the well-known inertia problem; on the other hand, we noted a jerky driving, making the experience of the human passenger not totally comfortable. A future direction of our research in this field may be the definition of a loss function that explicitly takes into account these negative effects on the driving task; currently, the inertia problem and the jerk are not considered in the adopted loss functions.


Training weather: Clear Noon
Result: Completed
Test weather: Soft Rain Sunset
Result: Completed
Training weather: Heavy Rain Noon
Result: Completed

(a) (b) (c)

Fig. 3. Example frames extracted from the CARLA simulator during the execution of an episode of our best model. **(a)** The vehicle is stopped at the red traffic light. **(b)** The vehicle starts when the traffic light is green. **(c)** The vehicle stops beyond a car.

5 Conclusions

In this work, we presented the first comprehensive, comparative analysis of the most common monomodal inputs that can be processed by a conditional imitation learning approach, namely CILRS [6]. In particular, we considered raw and processed data, namely RGB, depth and semantic segmentation images, as well as their multimodal combinations.

The achieved results showed that RGB outperformed the other monomodal inputs, in terms of success rate on the most popular benchmark NoCrash. However, RGB alone did not generalize well when tested on different weather conditions. This confirms the limitations of a monomodal approach and that an appropriate fusion of multimodal sensors is an effective approach in autonomous vehicle driving. The achieved overall results showed that the best configuration was RGBS compared to the others. Interestingly, the development of CIL-based models using different input data might offer reliable systems in the case of sensor fault (e.g., depth image acquisition devices), by deploying the best performing CIL approaches, trained on different input data, on board.

Future work will be devoted to the extension of the current CIL approach proposed by Codevilla *et al.* [5,6], by injecting the driving rules into the learning procedure and fully exploiting the full potential of imitation learning. In real-world applications, we aim at assessing the semantic segmentation obtained by a CNN-based solution, such as BiSeNet V2 [19].

References

1. Behl, A., Chitta, K., Prakash, A., Ohn-Bar, E., Geiger, A.: Label efficient visual abstractions for autonomous driving. In: Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 2338–2345. IEEE (2020)
2. Bojarski, M., et al.: End to end learning for self-driving cars. arXiv preprint arXiv:1604.07316 (2016)
3. Bojarski, M., et al.: Explaining how a deep neural network trained with end-to-end learning steers a car. arXiv preprint arXiv:1704.07911 (2017)
4. Chen, D., Zhou, B., Koltun, V., Krähenbühl, P.: Learning by cheating. In: Proceedings of Conference on Robot Learning, pp. 66–75. PMLR (2020)
5. Codevilla, F., Müller, M., López, A., Koltun, V., Dosovitskiy, A.: End-to-end driving via conditional imitation learning. In: IEEE International Conference on Robotics and Automation (ICRA), pp. 4693–4700. IEEE (2018)
6. Codevilla, F., Santana, E., López, A.M., Gaidon, A.: Exploring the limitations of behavior cloning for autonomous driving. In: Proceedings of IEEE/CVF International Conference on Computer Vision (ICCV), pp. 9329–9338 (2019)
7. Cultrera, L., Seidenari, L., Becattini, F., Pala, P., Del Bimbo, A.: Explaining autonomous driving by learning end-to-end visual attention. In: Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, pp. 340–341 (2020)
8. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: ImageNet: a large-scale hierarchical image database. In: Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 248–255. IEEE (2009)
9. Dosovitskiy, A., Ros, G., Codevilla, F., Lopez, A., Koltun, V.: Carla: an open urban driving simulator. In: Conference on Robot Learning, pp. 1–16. PMLR (2017)
10. Eraqi, H.M., Moustafa, M.N., Honer, J.: Efficient occupancy grid mapping and camera-lidar fusion for conditional imitation learning driving. In: Proceedings of IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC), pp. 1–7. IEEE (2020)
11. Holm, S.: A simple sequentially rejective multiple test procedure. Scand. J. Statist. **6**(2), 65–70 (1979). https://doi.org/10.2307/4615733
12. Huang, Z., Lv, C., Xing, Y., Wu, J.: Multi-modal sensor fusion-based deep neural network for end-to-end autonomous driving with scene understanding. IEEE Sensors J. MINO **21**(10), 11781–11790 (2020)
13. Ohn-Bar, E., Prakash, A., Behl, A., Chitta, K., Geiger, A.: Learning situational driving. In: Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 11296–11305 (2020)
14. Tampuu, A., Matiisen, T., Semikin, M., Fishman, D., Muhammad, N.: A survey of end-to-end driving: architectures and training methods. IEEE Trans. Neural Netw. Learn, Syst. (2020)
15. United States Department of Transportation: Risky Driving (2021). https://www.nhtsa.gov/. Accessed 18 Nov 2021

16. Wilcoxon, F.: Individual comparisons by ranking methods. Biometrics Bull. **1**(6), 196–202 (80–83). https://doi.org/10.2307/3001968
17. Xiao, Y., Codevilla, F., Gurram, A., Urfalioglu, O., López, A.M.: Multimodal end-to-end autonomous driving. IEEE Trans. Intell. Transp. Syst. (2020)
18. Xu, H., Gao, Y., Yu, F., Darrell, T.: End-to-end learning of driving models from large-scale video datasets. In: Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2174–2182 (2017)
19. Yu, C., Gao, C., Wang, J., Yu, G., Shen, C., Sang, N.: BiSeNet V2: Bilateral network with guided aggregation for real-time semantic segmentation. Int. J. Comput. Vis. **129**(11), 3051–3068 (2021)
20. Yurtsever, E., Lambert, J., Carballo, A., Takeda, K.: A survey of autonomous driving: common practices and emerging technologies. IEEE Access **8**, 58443–58469 (2020)