Effects of environmental feature selection on end-to-end vehicle steering controller

Dongjie Liu¹, Jin Zhao¹,², Zhuo Cao¹, Xinian Huang¹, Axing Xi³

¹School of Mechanical Engineering, Guizhou University, Guiyang, Guizhou, People’s Republic of China
²E-mail: zhaojin9485@163.com

Abstract: The method of deep learning has been widely used for end-to-end vehicle controllers’ training because of its approximately non-linear functions. However, the convolutional neural network (CNN) training process requires a large amount of labelled data sets and takes a lot of time. Aiming at this problem, three experimental frameworks are designed to study the influence of single features on end-to-end controller in the simple and complex environments. The performance of controllers trained with different features is analysed, and the criteria of feature selection are given to reduce calculation cost. Firstly, two types of images with different environmental complexities are collected and pre-processed into three types of missing feature sets (data sets for sky, roadside, and road features are discarded, respectively). Then, based on the NVIDIA network model, the feasibility and road verification of the three frameworks designed are carried out. The experimental results show that: (i) Road features are indispensable to the training controller in simple or complex environments; (ii) roadside features help to improve the generalisation of the controller; (iii) sky features play a limited role in training vehicle steering controllers.

1 Introduction

In recent years, with the rapid development of big data and artificial intelligence, deep learning has been widely used in image classification, natural language processing, target detection and motion modelling due to its powerful ability to approximate highly non-linear functions [1–4]. The convolutional neural network (CNN) reduces the dimensionality of high-dimensional inputs by convolution and is suitable for image recognition [5]. Therefore, the utilisation of CNN is widespread in the environment perception of autonomous vehicles [6], such as the classification of pedestrians and vehicles [7–9]. In the car driving assistance system, the identification of traffic signs and traffic police gestures is increasingly used by many universities domestically and abroad [10–13]. Self-driving cars using end-to-end learning, the DARPA Autonomous Vehicle (DAVE) and DAVE-2 in the DARPA project have been proven that end-to-end trained neural networks can indeed drive cars on public roads [14]. In [15], Muller et al. robot end vehicle controller in real time to achieve the obstacle detection and navigation. The literature [16] uses CNN to train end-to-end vehicle controllers to track curved lanes accurately. In [5], cars can be driven on standardised roads without lane markings and also on country lanes autonomously.

Compared to traditional lane keeping technology, autonomous vehicles based on end-to-end learning can eliminate steps such as lane marking detection, path planning and vehicle control. The end-to-end learning method does not need to detect and identify specific categories of predefined objects, nor does it need to label and match objects during training. Therefore, less manual work was required, which effectively solved the problem that the traditional method had low detection rate and poor real-time performance. However, [15, 17, 18] directly fed the original image into the neural network for training. Since the training CNN required a great number of data of various environments, it would take lots of time to process. Researchers used multiple graphics processing units (GPUs) to solve this problem [19], but this increased development costs because the GPU’s program structure is very different from the traditional central processing units, making GPU coding difficult to master. In [20], the training time was shortened by using low-resolution images, but this resulted in lower training accuracy. Shalev-Shwartz and Shashua [21] and Ohn-Bar and Trivedi, [22] sorted the object-level importance in the image by training a neural network with semantic abstraction or human-centered annotation. Inspired by Yang’s et al. [23], we further analyse the performance of end-to-end vehicle steering controller in complex environment, and give the criterion of feature selection.

We consider the effects of different complexity environments and different environmental features on training end-to-end vehicle steering controllers. Three frameworks are proposed to study the effects of single features on end-to-end vehicle steering controllers in simple and complex environments. Through simulation test and road test, this paper tries to balance the performance of controller with the cost of training time, and gives the reference criterion of feature selection, which can reduce the calculation cost without degrading the performance of steering controller.

Section 2 introduces the experimental design method. Section 3 analyses and evaluates the trained end-to-end vehicle controller. Simulation experiments and road experiments were carried out in Sections 4 and 5.

2 Methodology

We use the Udacity open source driverless simulator to collect two types of driving scene images with labels (steering angle and speed labels). One is used to train the end-to-end vehicle steering controller, and the other is used to verify the trained controller model (collectively referred to as the new data set). The two types of image data sets collected are pre-processed. Then, three different experimental frameworks were proposed to train and validate CNNs.

2.1 Data collection

The driving scene image with the steering angle mark was collected by the Udacity simulator. In order to ensure the consistency of the experiment, we selected the same road surface and surrounding environment as test scene, and collected data in four kinds of double-lane tracks, as shown in Fig. 1a. For the sake of minimising a large number of similar frames and improving the reliability of information, the image was sampled at 10 frames per second [5]. Around 8001 images with a size of 160 × 120 were collected. The steering angle and speed were normalised and the value range was [−1, 1]. Examples of road scene are shown in Fig. 1.
2.2 Improved NVIDIA network architecture

The first two fully connected layers in the NVIDIA network structure have no dropout layers [5]. To avoid overfitting, we improved performance by adding dropout layers to the network, with a Dropout ratio of 0.2. NVIDIA uses the activation function rectified linear unit (ReLU), but it has a disappearing gradient problem, so the exponential linear unit (ELU) activation function is used to overcome this problem. Compared with ReLU, ELU functions usually have a faster learning rate [24]. Finally, we have added regularisations to improve the generalisation of the network. The input layer is the pre-processed image, and the output layer is the predicted corner information of the vehicle. The first two layers of the convolutional layer use a $5 \times 5$ convolution kernel and a $2 \times 2$ step size, and the third and fourth convolution layers use a $3 \times 3$ convolution kernel and a $2 \times 2$ step size. The last convolutional layer uses a $3 \times 3$ convolution kernel and a $1 \times 1$ step size. Due to the feature maps are small, there is no pooling layer. The other parameters are batch_size = 128, epochs = 1000.

2.3 Verification of validity of CNN

We use two norms to verify that the improved NVIDIA network architecture is reasonable. That is, the training loss value and the performance of the controller on the new data set. Fig. 2a shows the training loss value, and the curve convergence can be found after 1000 iterations, so the training process can be stopped. The improved network model has significantly lower loss values than NVIDIA. Loss value convergence point is an important indicator for training neural networks, but the performance of trained end-to-end controllers on new data sets plays the most significant role. Fig. 2b shows the comparison of the predicted values of the two controllers with the true steering values on the new data set. It can be seen that the improved network is closer to the true value, while the original network is slightly biased. Therefore, it is proved that the improved neural network structure is reasonable, and the trained controller has a better performance.

2.4 Experimental framework design

2.4.1 Framework 1: In Framework 1, an end-to-end vehicle controller trained with 8001 original images is represented as CNN1. After that, CNN1 is tested with a new data set that discards certain features. We manually divide the features of the test set into three categories, namely sky features, roadside features, and road features. The feature area is shown in Fig. 2b. The sky feature (Zone A) refers to the upper part of the image, usually with clouds and buildings. Roadside features (Zone B) represent the left and right parts of the image, usually with grass, trees, and buildings. The road-related feature (Zone C) is the middle and lower part of the image, with a textured road. During the testing phase, each feature will be discarded one by one to assess its importance. The specific framework is shown in Fig. 3a.

2.4.2 Framework 2: In the Framework 2 shown in Fig. 3b, different features in the environment are discarded, and three different types of data sets are obtained, which are discarded sky-related features, roadside-related features and road-related features. Then, three types of new data sets are fed into the CNN, and three end-to-end vehicle steering controllers (CNN2, CNN3, and CNN4) are separately trained. Finally, three controllers are tested in the raw data set to evaluate the level of importance of individual features.

2.4.3 Framework 3: In order to study the influence of feature selection in complex environment on controller performance, another type of environment with relatively complex images is collected. The sample picture is shown in Fig. 4. Scenes 3 and 4 are more complex than the roadside environment compared to Fig. 1b. In practical applications, Framework 2 provides a direct reference with discarding the features. Therefore, the complex environmental data is also itemised into three categories (discarding sky features, roadside features, and road features). Then CNN2-1, CNN3-1, and CNN4-1 are trained, respectively. We make the comparison diagram between the predicted value and the real value of the controller on the new dataset, and compare the conclusions obtained in the simple environment. Finally, we combine the MAE, RMSE values and convolutional neural network visualisation results to analyse the impact of feature selection in complex environments on controller performance. The specific framework is shown in Fig. 3c.

3 Feature analysis and evaluation

Based on Frameworks 1 and 2, CNN2, CNN3, and CNN4 controllers are utilised to evaluate the importance of sky feature, roadside feature, and road feature in simple environment. Framework 3 investigates the effects of feature selection on end-to-end vehicle steering controllers in complex environments.

3.1 Feature analysis based on frame 1

In Framework 1, CNN1 is tested by a data set of discarded sky features, roadside features, and road features, respectively. It provides a direct understanding of the relationship between the predicted value of the end-to-end controller and the degree of importance of different features in the environment.

Fig. 5 shows the comparison of CNN1's predicted values on different discard feature sets with actual angle values. It can be clearly seen that the predicted value of CNN1 on the discarded sky
feature dataset is highly matched with the driver's steering value, indicating that the trained controller can control the vehicle without knowing the sky related features. Despite the predicted steering value using the discarded roadside feature dataset is slightly deviating from the driver's steering value, the trend of which is similar to the actual values. Accordingly, the roadside feature can be considered to provide the driver with some useful information. However, so as to obtain a more accurate driving command, the roadside features should not be discarded theoretically. The predicted value of the controller trained with the discarded road feature is larger than the actual value, which suggests that road features are essential to neural network learning and cannot be dismissed in a simple environment.

### 3.2 Feature evaluation based on Framework 2

In Framework 2, CNN2, CNN3, and CNN4 are tested separately using a data set containing all the features of the environment. This framework evaluates the impact of controllers trained with different drop feature data sets on vehicle steering performance.

Fig. 6 shows the comparison of the predicted values of the controller CNNs on the new data set with human driver steering values. Compared with Fig. 5, the three controllers are not good enough. CNN2 and CNN3 have better performance, and CNN4 has the worst performance, completely deviating from the actual value. So in the simple environment, the lack of road features causes the controller to malfunction. Table 1 shows that the MAE, RMSE are between the predicted value of CNNs on the new data set and labelled value. Controller CNN2 has the smallest MAE value. The MAE and RMSE values of CNN4 are 0.1924 and 0.2363, respectively, which is an order of magnitude higher than CNN2 and CNN3. Therefore, it can be inferred that road characteristics are essential for end-to-end learning in a simple environment. In contrast, sky-related features are the least important in end-to-end learning. The roadside feature can increase the generalisation ability of neural network learning, and provide assistance for obtaining accurate steering commands, but the significance of that...
is lower than the road features. Thus, in some unimportant and simple situations, sky and roadside features can be discarded to reduce time costs.

3.3 Feature evaluation based on Framework 3

In order to further analyse the feature selection criteria of end-to-end vehicle controller in complex environment, we collect complex environmental data sets. Discard the sky, roadside, and road features one by one and train the controllers separately. Finally, the performance of the controllers CNN2-1, CNN3-1, and CNN4-1 were tested by using a data set containing all environmental features.

Fig. 7 compares the actual values with the predicted values by CNN1-1 (controller under full feature) and CNN2-1, respectively. It can be seen that the performance of the two is almost the same, also they are consistent with the conclusions obtained in a simple environment: the sky feature is the least important for end-to-end learning. Fig. 7 reveals that CNN4-1, discarding road features, performs greatly, which is significantly better than CNN4 with missing road features in a simple environment. In a complex environment, the first layer of the network automatically extracts a large amount of information. After the fifth layer is convolved, the neural network extracts only the lane lines and a small number of roadside features. In a complex environment, the performance of the controllers CNN2-1 and CNN4-1 with complex roadside features are lower than the MAE and RMSE values of CNN2 and CNN4 in the simple environment. The MAE and RMSE values of the controller CNN-3 with less complex roadside features are much lower than the MAE and RMSE values of CNN3. Different from convolution results in simple environments (road features dominate): road features only extract lane line information, while roadside features in complex environments are extracted in large quantities and can be referenced. Therefore, it can be concluded that: (i) The complex roadside environment can greatly improve the performance of the end-to-end steering controller. (ii) In complex environments, roadside features are of remarkable importance, and still sky features are attached the least significance to analyse.

4 Simulation

To further analyse the impact of discarding related features on CNNs in simple and complex environments, two simple scenarios are used in the simulator (Fig. 4a), also another two complex scenarios (Fig. 4b) verify the feasibility of the proposed framework and the accuracy of the conclusions obtained above. Tables 3 and 4 show the successful lane-keeping time in simple and complex test scenarios. Combining the two tables, it can be

| Table 1 MAE and RMSE between CNNs predicted results and labelled values |
|-------------------|-------------------|-------------------|
|                    | CNN2              | CNN3              | CNN4              |
| MAE                | 3.38 × 10⁻²       | 7.51 × 10⁻²       | 1.924 × 10⁻¹      |
| RMSE               | 4.44 × 10⁻²       | 9.95 × 10⁻²       | 2.363 × 10⁻¹      |

| Table 2 MAE and RMSE between CNNs predicted results and labelled values |
|-------------------|-------------------|-------------------|
|                    | CNN2–1            | CNN3–1            | CNN4–1            |
| MAE                | 1.30 × 10⁻²       | 1.201 × 10⁻¹      | 3.01 × 10⁻²       |
| RMSE               | 1.98 × 10⁻²       | 1.588 × 10⁻¹      | 4.60 × 10⁻²       |
found that CNN2 in simple scenes or CNN2-1 in complex scenes can complete lane keeping, which proves that the end-to-end controller has the lowest dependence on sky features. Table 3 demonstrates that the performance of CNN1 and CNN2 is the greatest in scenarios 1 and 2, besides, CNN3 and CNN4 fail to complete lane-keeping. Since the CNN2 with both road and roadside features has longer lane-keeping time than the CNN3 which discards the roadside features. Moreover, CNN4, which discards road features in a simple scene, deviates from the lane in ~6 s. Therefore, it can be inferred that under a simple condition, what plays the most crucial role is the lane features, and the less important one are roadside features. Also the controller that discards the roadside features, CNN3 and CNN3-1 can only be kept for a while. The CNN3-1 lane keeping time in the complex environment is shorter than CNN2-1 and CNN4-1. The controller CNN2-1, including includes both road and roadside features, successfully achieved lane keeping. Although controllers discard road features, the success rate of CNN4-1 lane keeping in complex scene is much higher than that of CNN4 in simple scene. In the simple scene, the success rate of lane keeping from CNN4 discarding the road feature is lower than that of CNN3, in which operational outcome does not exhibit in the complex scene. As a result, it can be concluded that: (i) higher environmental complexity can improve the performance of the end-to-end steering controller. (ii) In complex environments, roadside features should not be discarded, in addition, sky features remain so ordinary that can be discarded to accelerate training speed of neural network.

5 Road experiment

For verifying the feasibility of the proposed three frameworks and the correctness of the simulation conclusions, a road test platform was built to conduct experiments. Fig. 9a shows a simple environment and Fig. 9b reveals a complex environment. Different objects and different types of paper sheets are placed on the roadside to simulate complex environments. By comparing Bluetooth, WiFi, DSRC, and ZigBee performance in short-range wireless communication, the smart car driving system uses a 2.4 GHz WiFi band. The hardware and software platform is established as shown in Fig. 9c, and the road experiment is carried out by using WiFi to realise communication between workstation, intelligent vehicle, X-Box and smart phone, driving 50 rounds autonomously in the experimental field, and testing 2 rounds per round. The sign of successful autonomous driving is that the car does not run out of the track and does not hit the fence, otherwise it will fail.

### Table 3

|            | CNN1 | CNN2 | CNN3, s | CNN4, s |
|------------|------|------|---------|---------|
| scene 1    | finish (310 s) | Finish (318 s) | 237      | 4       |
| scene 2    | finish (87 s)  | Finish (94 s)  | 31       | 7       |

### Table 4

|            | CNN1-1 | CNN2-1 | CNN3-1, s | CNN4-1, s |
|------------|--------|--------|-----------|-----------|
| scene 3    | finish (148 s) | finish (153 s) | 69        | 101       |
| scene 4    | finish (344 s) | finish (350 s) | 163       | 229       |

### Table 5

|                      | CNN1, % | CNN2, % | CNN3, % | CNN4, % |
|----------------------|---------|---------|---------|---------|
| success rate         | 90.0    | 88.0    | 62.0    | 18.0    |

### Table 6

|                      | CNN1-1, % | CNN2-1, % | CNN3-1, % | CNN4-1, % |
|----------------------|-----------|-----------|-----------|-----------|
| success rate         | 96.0      | 96.0      | 64.0      | 82.0      |

From Tables 5 and 6, conclusions can be made that in a simple or complex environment, there is barely distinction between the controllers whether to discard the sky feature from full features. No matter how complicated the environment is, CNN3 and CNN4, CNN3-1 and CNN4-1 have lower success rates than controllers that discard sky features. Therefore, sky feature is the least important in the end-to-end learning. In a simple environment, CNN3 runs much better than CNN4, and controllers that discard road features are almost inoperable. It can be seen that in a simple environment, the end-to-end controller has the highest dependence on road features and the roadside features rely less on them. The controller CNN4 in a simple environment runs much lower than CNN4-1. Despite the lack of road-related features, the controller can be assisted and the controller still performs well, due to being in the complex environment. In complex environment, regardless of the sky or road features, the results are better than the controller in simple environment. In particular, the full features controller CNN1-1 runs at 96.0%, which is higher than that of CNN1. Consequently, the end-to-end controller performs better in a complex environment. Both the CNN3-1 with missing roadside feature and the CNN4-1 with missing road feature are inferior to the controller CNN2-1 with two features. Therefore, if you want to improve the reliability of the end-to-end steering controller in complex environments, roadside features cannot be discarded in end-to-end learning, and sky features can be discarded to improve neural network training speed.

From Tables 5 and 6, conclusion can be made that in a simple or complex environment, the controller that discards the sky feature is almost similar to the controller with full features. Whether it is a simple or complex environment, CNN3 and CNN4, CNN3-1 and CNN4-1 have lower success rates than controllers that discard sky features. Hence, sky feature is the least important in end-to-end learning. In a simple environment, CNN3 runs much better than CNN4, and controllers that discard road features are almost inoperable. Therefore, the dependence of end-to-end controller on the road features are the highest and the most significant, moreover, the roadside features take the second place in a simple environment. The controller CNN4 in a simple environment runs much lower than CNN4-1. Despite the lack of road-related features, the controller can be assisted and the
controller still performs well, due to being in the complex environment. In complex environment, regardless of the sky or road features, the results are better than the controller in simple environment. In particular, the full features controller CNN1-1 runs at 96.0%, which is higher than CNN1. So the end-to-end controller performs better in a complex environment. Both the CNN3-1 with missing roadside feature and the CNN4-1 with missing road feature are inferior to the controller CNN2-1 with two features. Therefore, if you want to improve the reliability of the end-to-end steering controller in complex environments, roadside features cannot be discarded in end-to-end learning, and sky features can be discarded to improve neural network training speed.

6 Conclusion

In this paper, some features of the image are discarded for neural network learning, which solves the problem that a large number of data sets need to be marked in the training process and the time cost is high. It is conceivable that CNN does not need to traverse the entire image when doing convolution operations, thus reducing the amount of computation. However, the importance of the three features is unknown, and the performance of the controller trained by arbitrarily discarding related features will not be guaranteed. Therefore, three frameworks were designed to study the effects of single features on the performance of end-to-end vehicle controllers in simple and complex environments, and feature selection criteria were given to reduce computational costs. In a simple environment, the effects of controllers trained with different discard feature sets on vehicle steering were evaluated based on frames 1 and 2, which provides a direct understanding of the time cost and performance trade-off of controller and the degree of importance of different features in the complex environment. Based on the Framework 2, the Framework 3 analyses the influence of the selection of features in the complex environment on the controller. However, the main contribution of this paper is to supply a framework for feature analysis and selection for end-to-end learning. This framework can provide reference for the time cost and performance trade-off of controller training based on deep learning theory. In this paper, we only study specific scenarios in simple and complex environments. The future will also consider the impact of weather conditions, traffic conditions and roadside environmental changes on experimental results.

7 Acknowledgments

This work was supported by The Outstanding Young Researcher of Guizhou Province [2017]5630 and Technology Research and Development Program of Guizhou Province [2017]2027 and [2018]2168.

8 References

[1] LeCun, Y., Bengio, Y., Hinton, G.: ‘Deep learning’, Nature, 2015, 521, (7553), pp. 436–444
[2] Gu, J., Wang, Z., Kuen, J., et al.: ‘Recent advances in convolutional neural networks’, Pattern Recognit, 2018, 79, pp. 354–377
[3] Conneau, A., Schwenk, H., Barrault, L., et al.: ‘Very deep convolutional networks for natural language processing’, arXiv preprint: 1606.07181, 2016
[4] Shao, L., Cai, Z., Liu, L., et al.: ‘Performance evaluation of deep feature learning for RGB-D image/video classification’, Inf. Sci., 2017, 385, pp. 266–283
[5] Bojarski, M., Del Testa, D., Dworakowski, D., et al.: ‘End to end learning for self-driving cars’. arXiv preprint arXiv:1604.07316, 2016
[6] Chen, C., Seff, A., Vertanen, A., et al.: ‘Deepdriving: learning affordance for direct perception in autonomous driving’, Proc. Int. Conf. Int. Conf. on Computer Vision, Santiago, Chile, 2015, pp. 2722–2730
[7] Ishak, K.A., Samad, S.A., Hussain, A.: ‘A face detection and recognition system for intelligent vehicles’, Inf. Technol. J., 2006, 5, (3), pp. 507–515
[8] Tian, Y., Luo, P., Wang, X., et al.: ‘Pedestrian detection aided by deep learning semantic tasks’. IEEE Int. Conf. Conf. on Computer Vision and Pattern Recognition, Boston, MA, USA, June 2015, pp. 5079–5087
[9] Xu, D., El-Khamy, M., Lee, J., et al.: ‘Fused DNN: a deep neural network fusion approach to fast and robust pedestrian detection’. IEEE Int. Conf. Conf. on Applications of Computer Vision (WACY), Santa Rosa, CA, USA, March 2017, pp. 953–961
[10] CireşAn, D., Meier, U., Masci, J., et al.: ‘Multi-column deep neural network for traffic sign classification’, Neural Netw., 2012, 32, pp. 333–338
[11] Stallkamp, J., Schlippsing, M., Salmen, J., et al.: ‘Man vs. computer: benchmarking machine learning algorithms for traffic sign recognition’, Neural Netw., 2012, 32, pp. 323–332
[12] Zhu, G., Zhang, L., Shen, X., et al.: ‘Multimodal gesture recognition using 3-D convolution and convolutional LSTM’. IEEE Access., 2017, 5, pp. 4517–4524
[13] Le, Q.K., Pham, C.H., Le, T.H.: ‘Road traffic control gesture recognition using depth images’, IEEE Trans. Smart. Process. Comput., 2012, 1, (1), pp. 1–7
[14] LeCun, Y., Cosatto, E., Ben, J., et al.: ‘DAVE: autonomous off-road vehicle control using end-to-end learning’. DARPA-IPTO Final Report, 2004
[15] Muller, U., Ben, J., Cosatto, E., et al.: ‘Off-road obstacle avoidance through end-to-end learning’. Proc. Int. Conf. Advances in Neural Information Processing Systems, Vancouver BC, Canada, 2006, pp. 739–746
[16] Xu, H., Gao, Y., Yu, F., et al.: ‘End-to-end learning of driving models from large-scale video datasets’. Proc. Int. Conf. Computer Vision and Pattern Recognition, Honolulu, USA, June 2015, pp. 2181–2188
[17] Chen, Z., Huang, X.: ‘End-to-end learning for lane keeping of self-driving cars’. IEEE Int. Conf. 2017 IEEE Intelligent Vehicles Symp. (IV), Redondo Beach, CA, USA, June 2017, pp. 1856–1860
[18] Agarwal, G., Räber, L., Shen, Y., et al.: ‘End-to-end deep imitation learning: robot soccer case study’. arXiv preprint arXiv:1807.09205, 2018
[19] Simonoyan, K., Zisserman, A.: ‘Very deep convolutional networks for large-scale image recognition’. arXiv preprint arXiv:1409.1556, 2014
[20] Li, H., Lin, Z., Shen, X., et al.: ‘A convolutional neural network cascade for face detection’. Proc. Int. Conf. Computer Vision and Pattern Recognition, Boston, MA, USA, June 2015, pp. 5325–5334
[21] Shalev-Shwartz, S., Shashua, A.: ‘On the sample complexity of end-to-end training vs. semantic abstraction training’. arXiv preprint arXiv:1604.06915, 2016
[22] Ohn-Bar, E., Trivedi, M.M.: ‘Are all objects equal? Deep spatio-temporal importance prediction in driving videos’, Pattern Recognit., 2017, 64, pp. 425–436
[23] Yang, S., Wang, W., Liu, C., et al.: ‘Feature analysis and selection for training an end-to-end autonomous vehicle controller using deep learning approach’. IEEE Int. Conf. 2017 IEEE Intelligent Vehicles Symp. (IV), Redondo Beach, CA, USA, June 2017, pp. 1033–1038
[24] Clevert, D.A., Unterthiner, T., Hochreiter, S.: ‘Fast and accurate deep network learning by exponential linear units (elu)’. arXiv preprint arXiv:1511.07289, 2015