Thorough Analysis and Reasoning of Environmental Factors on End-to-End Driving in Pedestrian Zones

Qazi Hamza Jan, Arshil Ali Khan and Karsten Berns

RPTU Kaiserslautern-Landau, Erwin-Schrödinger-Straße 52, 67663, Kaiserslautern, Germany

Keywords: Autonomous Driving, Pedestrian Zones, End-to-End Driving.

Abstract: With the development of machine learning techniques and increase in their precision, they are used in different aspects of autonomous driving. One application is end-to-end driving. This approach directly takes in the sensor data and outputs the control value of the vehicle. End-to-end systems have widely been used. The goal of this work is to investigate the effect of change in weather condition, presence of pedestrians, and reason the prediction failure, along with improving the results in a pedestrian zone. Driving through the pedestrian zone is challenging due to the narrow path and crowd of people. This work uses RGB images from a front-facing camera mounted on the roof of a minibus and outputs the steering angle of the vehicle. A Convolutional Neural Network (CNN) is implemented for regression prediction. The testing was first done in a simulation environment which comprised of the replicated version of the campus, the sensor system and the vehicle model. Thorough testing is done in different weather conditions and with the simulated pedestrians to check the robustness of the system for such diversified changes in the environment. The vehicle avoided the simulated pedestrians placed randomly at the boundary of narrow paths. In an unseen environment, the vehicle approached the region with the same texture it was trained on. Later, the system was transferred to a real machine and further trained and tested. Due to unavailability of the ground truth, the results can not be delineated for real world testing, but are reasoned through visual monitoring. The vehicle followed the path and performed well in an unseen environment as well.

1 INTRODUCTION

The automotive industry is increasing the autonomy of their vehicles for a better drive experience. This rapid evolution of current automotive technology has the goal to deliver greater safety benefits and a variety of autonomous driving systems (Levinson et al., 2011). The fundamental concept of autonomous driving is to have a sensor reading from different sensors and input into a Driving Decision Making (DDM) system. The vehicle is driven based on the control approach implemented in the DDM. Two main approaches for autonomous driving are modular and end-to-end. In the modular approach, the navigation, path-planning, safety, etc., are separately done (Yurtsever et al., 2020). Modular systems can be easily identified and are replaceable, but it becomes costly to maintain them. End-to-end systems, on the other hand, uses the pre-processed data directly from the sensors fed into the model and gives the relevant output such as steering, velocity, or braking of the vehicle. Such models are supervised by training with huge data (Argall et al., 2009) or by improving the results based on reward function with reinforcement learning (Sutton et al., 1998).

Figure 1: Driverless Minibus on campus of RPTU Kaiserslautern-Landau (Jan and Berns, 2021). It is meant to ferry 6-8 people from building to building in the campus.

Autonomous vehicles (AVs) have commenced operating in Pedestrian zones as well, to provide means of transport for, especially elderly and disabled peo-

Jan, Q., Khan, A. and Berns, K.
Thorough Analysis and Reasoning of Environmental Factors on End-to-End Driving in Pedestrian Zones.
DOI: 10.5220/0012242900003543
In *Proceedings of the 20th International Conference on Informatics in Control, Automation and Robotics (ICINCO 2023) - Volume 1*, pages 495-502
ISBN: 978-989-758-670-5; ISSN: 2184-2809

ple. This is due to the increasing lengths of pedestrian zones. Such AVs mainly assist traveling from building to building which also fits in the category of first and last-mile travel. Driving through pedestrian zones offers major challenges. Unlike ordinary street view, it has unstructured paths without markings and highly dynamic obstacles - the pedestrians themselves. Having said that, these pedestrians cross the vehicle while remaining on the boundary of the path. Unpredictable crossing decision from pedestrians impels the vehicle to drive irregularly. For DDM systems using a modular approach, it becomes difficult to navigate precisely in narrow paths due to highly imprecise GPS signals in a closed and cluttered building area (Chang et al., 2009). The local mapping is also frequently updated as a consequence of recurring pedestrians passing by the vehicles. To remove such interruptions, researchers are relying more on interacting with the pedestrians (Jan et al., 2020b). Although it reduces the braking behavior of the vehicle by giving the vehicle's intent in advance, it does not stop the pedestrians from crossing the vehicle.

With the aforementioned motivation and challenges, this paper focuses on exploring the usage of an End-to-end system for an AV in a pedestrian zone. It uses a CNN which feeds on RGB images to predict steering values. A detailed investigation is done to see the effect of unstructured environment, different weather conditions, and presence of pedestrians. Due to unknown results at the start of the work and safety concerns of the pedestrians, initial testing was done in the simulation environment. Apart from hardware troubles, simulation offers to recreate a dedicated testing scenario which aligns with the idea of this work.

The final goal of this work was to bridge a gap between the simulated to a real vehicle shown in figure 1. The vehicle is configured with safety certified system (Jan and Berns, 2021) to avoid any kind of collision in case of incorrect prediction.

The novelty of this work is summarized below:

- Preparation of training data.
- Impact of unstructured crossings and inconsistent width lanes on End-to-end learning.
- Effect of different weather conditions, such as sunny, foggy, rainy and snowy on the driving behavior.
- Finding the change in driving pattern in the presence of a pedestrian.
- Bridging a gap between simulation and real-world systems.

After the related work in the next section, implementation details are given in the section 4. Extensive experiments and their evaluations for simulated environment are done in section 4.1. The transfer learning for real system is given in section 5.

2 RELATED WORK

The first use of a similar system is done in (Pomerleau, 1988), where the authors have used a CNN to predict 45 direction outputs with 29 hidden layers. Monocular camera and radar images were used as input to the network.

The authors in (Bojarski et al., 2016) did similar work, by training a CNN on the images captured. But instead of using a single camera, they used three cameras: the front facing camera, right camera and the left camera. The data was collected on a road scenario with different lighting conditions. NVIDIA's CNN architecture is used, where the network's weights were trained to reduce the mean squared error between the steering command predicted by the network and the steering angle by a Human driver. But before feeding the data to the network, data augmentation was done by adjusting random shift and rotation. The results were tested on a simulator.

End-to-end learning is also integrated with probabilistic algorithms to improve predicted steering angles (Hubschneider et al., 2017). They use a planning algorithm based on probabilistic sampling. Additional safety is provided by using end-to-end learning. This combination overcomes the black-box concept and provides a control instance for the output. In order to further improve the safety, trajectory optimization is performed which minimizes the given cost function, which in addition also helps in dynamic obstacles detection using some sensors.

Authors in (Lee and Ha, 2020) have also used image based end-to-end driving for autonomous vehicles. They have used a Long-term Recurrent Convolutional Network. They rely more on time-series vision data. They have experimented with their system in a simulator with a typical street like environment.

Similar work has been done by many researchers to use end-to-end driving systems on autonomous vehicles. Mostly, the work is done in simulation and few have tested it on a real vehicle. No work is found to have thoroughly tested the system for different factors like weather conditions, driving among pedestrians, and arbitrary structure of a pedestrian zone. This paper analyzes the effect of such factors.

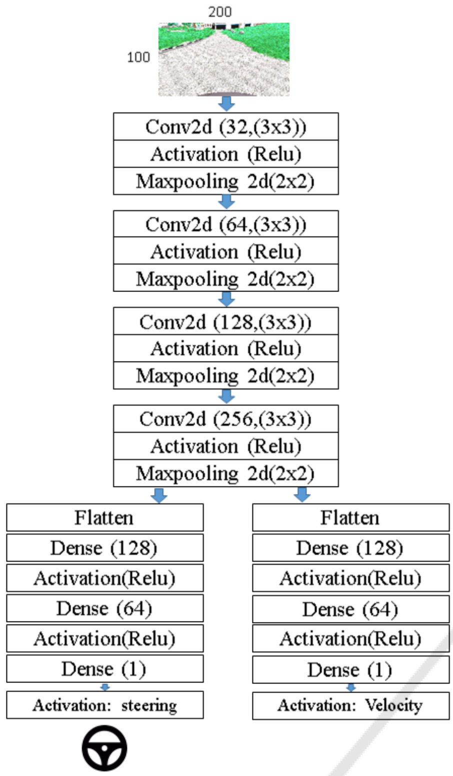


Figure 2: Network architecture.

3 NETWORK ARCHITECTURE

CNNs are extensively used in various applications (He et al., 2016; He et al., 2017; Chen et al., 2018). Such networks simplifies the process of extracting features manually and predicting the corresponding outcome of the system. Detection and identification systems are easily implementable and can be tested on real hardware. For the training and testing purpose, a Keras Multi-Output CNN-Regression model was developed, where Keras functional API (Manaswi, 2018) is used to build a multi-output deep learning model.

After training the model with RGB images, the model was able to predict the two most important driving parameters: steering angle and the vehicle velocity. The output value of steering angle resides between -1 to +1 (normalization for maximum steering angle of the vehicle) and velocity values defined between 0 to 1 (normalization of maximum velocity of the vehicle). Hence, the model is named Multi-output CNN-Regression model.

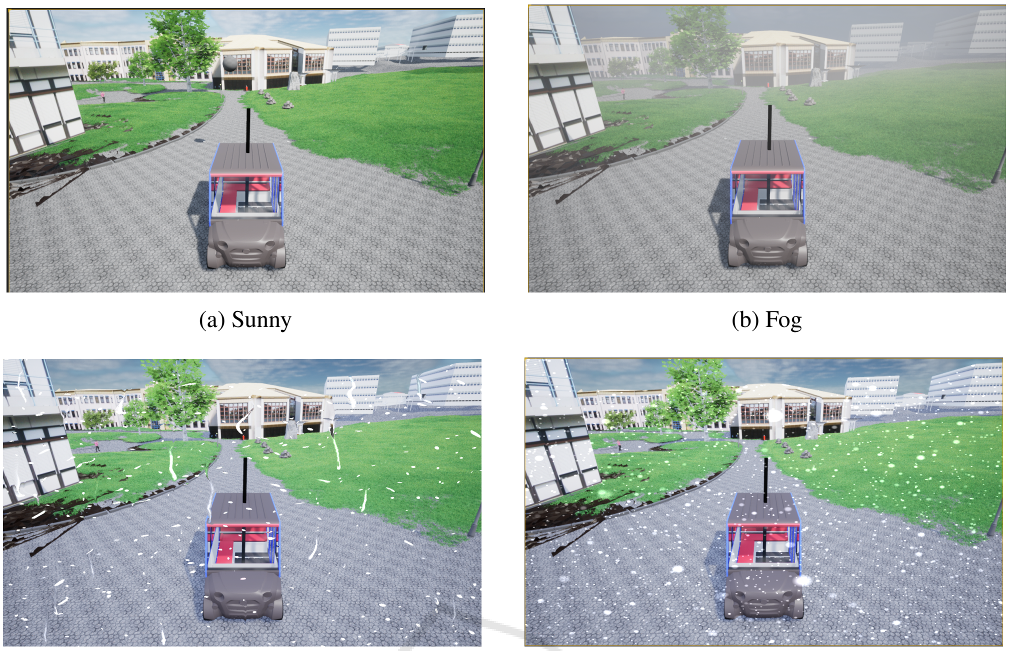
To determine the driving behavior by the network, for instance following the path and avoiding pedestrians, the prime concern was the steering of the vehicle; hence, only the steering node is considered and taken into use for this work. Meanwhile, velocity was not pertinent for these tests due to the following reasons: not to entangle the predicted steering values with varying values of velocity; low level safety modules override the velocity values due to safety concerns; speed did not have any contribution in the analysis of the result; it is not practical to control velocity in the crowded environment; and early testing did not show any discernible results. Therefore, the velocity was not taken into account and vehicle drove at a constant walking speed, i.e., 6km/h.

Initially, the input data, RGB image, resized to 200x100, is passed through the convolutional layers, i.e., through a CNN shown in figure 2, where the feature extraction is performed on the image data through multiple convolutions, max-pooling layers using Relu as the activation function. Once the features are extracted, regression layers are applied to the network to get the steering and velocity as the output.

4 SIMULATION

Since neural networks offer no satisfactory explanation for their outcome, it necessitates to begin testing in simulated environments. The campus of University of Kaiserslautern-Landau and the ego-vehicle shown in figure 1 were replicated in Unreal Engine (UE). UE has the advantage of fine rendering of the scene. This aids in realistic data of the visual sensor implemented in the simulation. It also includes a class to define the physics of the vehicle. The vehicle used for this work consists of a double ackermann steering to cater sharp turns. The interface (Wolf et al., 2020) between our robotic framework and UE is implemented to receive the camera images and, in return, control the vehicle. The interface provides RGB images from the sensor plugin in the UE which is fed to the network after resizing. The steering values from the network are denormalized to -22° to 22° (left to right) for the UE model of the vehicle. The established interface between UE and our framework is designed in such a manner to enable directly switching between real hardware and simulation.

Weather. In the UE editor, the sky-plugin is inserted which furnishes the environment with the possibility of weather conditions. Effects and changes in various elements of weather, for example, sun brightness, are available by tweaking the suitable parameters. The data was collected in a variety of conditions to test the network for every class of weather. The included classes were: sunny, rain, snow, and fog with different daylight conditions. Sample images of weather classes from the simulation can be seen in figure 3.



(a) Sunny (b) Fog

(c) Rain (d) Snow

Figure 3: Sample of different weather condition in the simulated environment.

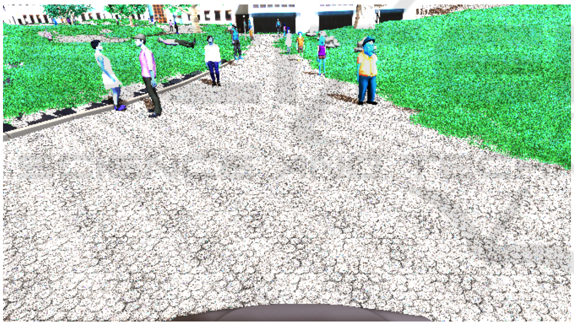


Figure 4: Data set sample for training images with pedestrians along the path in the campus.

Pedestrians. As the driving for this work was focused on the pedestrian zone, it was rational to drive through pedestrians in simulation and investigate how well the network performs. Simulated pedestrians are spawned to test the network in a crowded environment. The virtual pedestrians were used from our previous work (Jan et al., 2021; Jan et al., 2020a). These virtual pedestrians offer realistic behavior in terms of crossing the vehicle and reaching its goal. It was proposed to keep the pedestrians static on the boundary of the path, example shown in figure 4, to scrutinize the evasion behavior along with changing weather. This minimizes the testing variables.

Training Process. Neural networks for imitation learning heavily relies on how good training data is created. Hence, it becomes critical to make sure of the type of data and its distribution. Since there is no data set available which provides testing for situations like: change in weather, random placement of pedestrians, and irregular pathways in shared spaces, we had to generate our own data set. To prepare data for training, the simulated bus was driven by several users with a joystick on the different paths of the campus. RGB images from the front camera mount on the roof of the vehicle were recorded along with the steering values given by the user. The user was told to stay in between the path while driving. In case of pedestrian presences, the user was told to avoid them but not drive off the path.

Multiple pedestrians were randomly positioned along the border of the path because people are expected to take partial responsibility and give way to the vehicle. Some sample training images are given with the experiments. For every training sequence, the pedestrians were relocated and reoriented at different positions. Individual and group formation was incorporated. To measure how well the driving takes place in the test phase, a spline is created along the center of the drivable path. Spline assisted user to remain in path during training and and compare results during the test phase. Once the model is developed, training is started with the input image data as well as the ground truth values of steering angle. As mentioned in the previous section, the velocity was kept constant. The data augmentation is performed con-

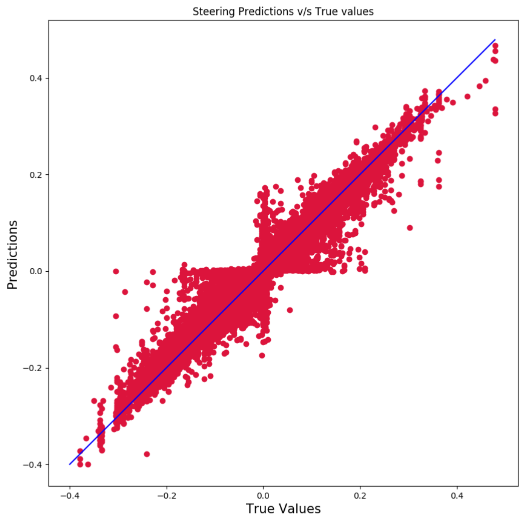


Figure 5: Scatter plot for Steering angle prediction v/s Ground Truth.

sisting of adding noise, flipping images along with the steering angles, and changing brightness. Furthermore, the data is split into 70 by 30 for the evaluation process.

For training, the images were collected at 15 fps. The vehicle was driven multiple times and data was recorded for all weather conditions and random pedestrian placement. The number of data collected during training is shown in the table 1. To validate the results, a scatter plot between predicted and ground truth is drawn in figure 5. For low true values, the model under steers. On the other hand, for zero steering the model has high noise. Possibly, high frequency of such low values were unnoticeable during driving.

4.1 Experiment and Evaluation

This section gives a detailed explanation of how different weather conditions and the presence of pedestrians affects the output of the network. Unseen data was also tested to comprehend the adaptation of the network. As explained in section 4, the simulated pedestrians were placed on the path to observe the slight change in steering while crossing them. Due to the restricted length of the paper, the results are shown

Table 1: Count of data collection for all classes.

Condition	Sunny	Snow	fog	Rain
Without Pedestrians	22250	10020	13796	13902
With Pedestrians	1801	3333	2605	2801

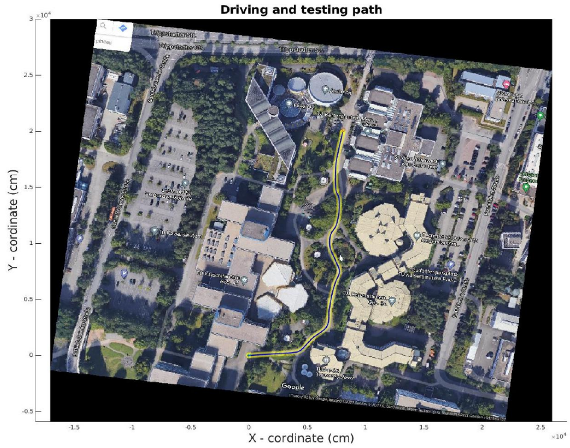


Figure 6: The image shows the structure of training and testing data at campus of RPTU Kaiserslautern-Landau. The model of campus was replicated in the model and same path was driven in the simulation. The path in the plots shown for testing is overlayed in black with yellow outline for coherence.

for the trajectory identified as yellow in figure 6. For all the graphs shown in this section, Blue is the spline trajectory in the center of the path as ground truth. Green is the predicted trajectory by the network. Red-circle represents the location of the collision and resetting the vehicle manually on the path.

The testing in the simulation was done in a fixed order to observe the improvement in results; the network was trained on one class and tested on the other class. In this way, we could identify the reasons for failures and dependency of one weather condition on another.

Sunny. The training was done with bright sunlight at different times of the day. The sample of weather conditions can be seen in figure 3a. Strong shadows are expected for this case which moves along the path of the sun. The graph in figure 7 shows the driving path. It can be seen for the sunny case that the collision happens at one point. Figure 8 shows the camera view before the collision point. This validates the fact that no matter how much training is done, the network, sometimes, fails to predict correctly in strong shadows.

Fog. To exploit the effect of variations in the simulation, network trained for sunny data was tested in fog. The left pathway in figure 9 shows that the vehicle collided at different places. After analyzing the image information at the collision points, one shown in figure 10, it is observed that the contrast level tremendously reduces, especially for green patches in the images. After training for fog data as well, no collision was seen in the test shown on the right of figure 9. Since no shadow exists in a foggy environment, the

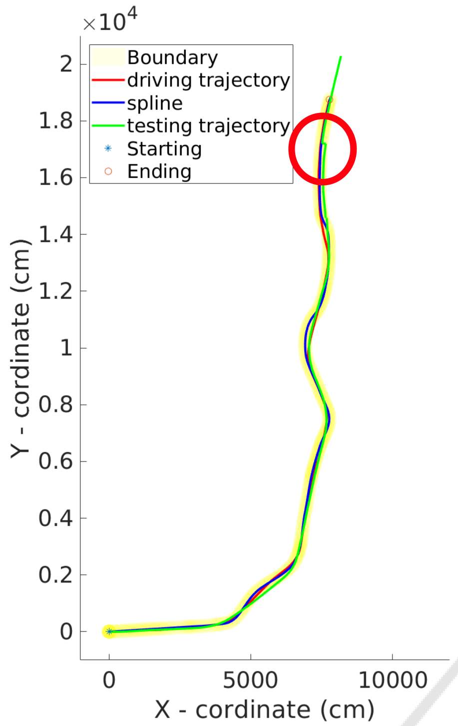


Figure 7: Testing in sunny environment in simulation. Red cicle show the collision due to strong shadows shown in figure 8.

Figure 8: Strong shadows during sunny condition.

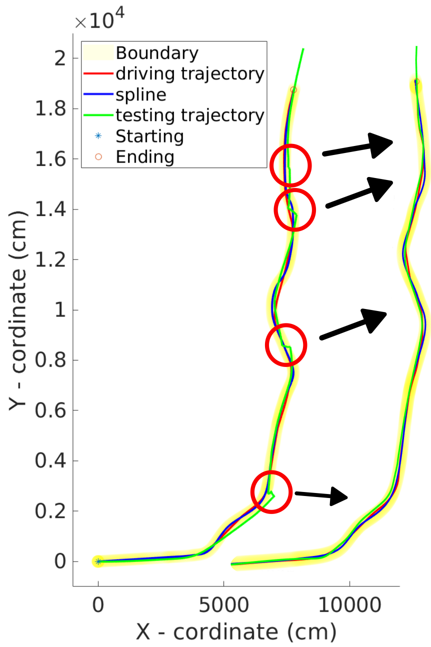


Figure 9: Testing in fog. Left path is trained for sunny class only with collision circled in red, whereas right path is trained for fog which avoids all the collisions.

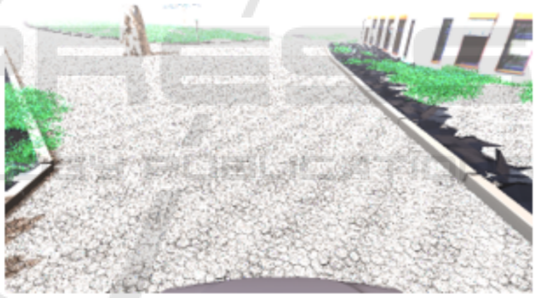


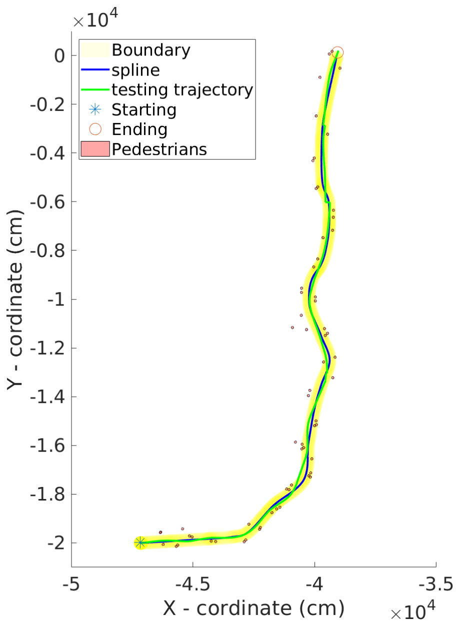
Figure 10: Effect of fog in the camera view.

collision happening in the sunny environment, due to shadows, is circumvented.

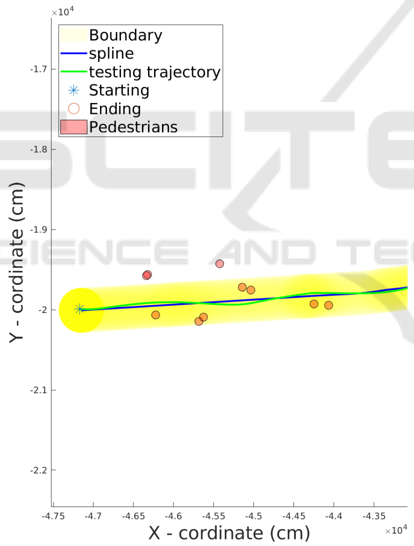
All. Similar tests were also performed for dusk, rain, and snow. The network drove without collision in rain and snow after being trained for dusk only. This establishes the fact that, mainly, the effect was the brightness of the scene and not the interference due to raindrops or snowflakes.

Pedestrians. After spawning pedestrians in the environment as shown in figure 4, the driving for training was done in such a way to sharply steer within the path to avoid collision. Training for pedestrians was done several times with different locations, orientations and number of pedestrians in a group. Here the focus was to see whether the vehicle steers to avoid the pedestrians and carefully pass by. Path including pedestrians is shown in figure 11a. figure 11b is a zoomed version of starting point shown in figure 11a. The predicted trajectory shown in green forms small curvatures to avoid pedestrians, which did not exist in experiments without pedestrians.

Unseen. For unseen data, the vehicle reacted differently. One particular case is shown in figure 12 where the vehicle is allowed to drive outside campus with different textures. The vehicle follows the red line shown in the figure 12. Since the system has only learned to drive on the texture shown in figure 3, the vehicle tries to follow the same texture instead of going straight on the dark texture. This asserts the importance of texture on which the network is trained on.



(a) Driving plot through pedestrians



(b) Zoomed area of the starting point in the above plot.

Figure 11: Driving through pedestrians in the simulation. (a) shows the pedestrian placement along the path, and (b) is a zoomed version from the starting point to show the change in driving prediction because of pedestrians.

5 REAL-WORLD

For the final work of this research, the objective was to test the model on a real vehicle. After thorough testing and successful driving in the simulation, it was now suitable to test the system for a real vehicle.

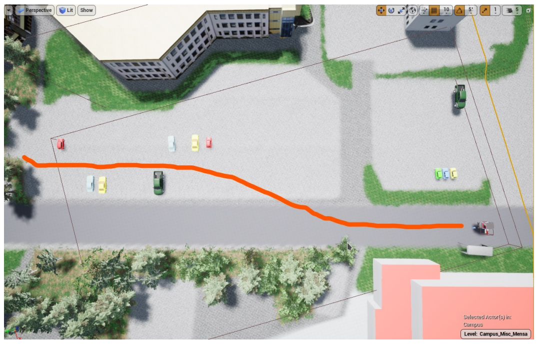


Figure 12: Testing in unseen and different texture in simulation. Red line shows the driven trajectory.

Figure 13: A comparison image in actual campus with the simulation shown in figure 9, but in sunny condition.

Figure 14: Testing the system on real unseen environment.

Although the campus, sensor system, and vehicle model were replicated in the simulation, there always exists a gap between the real and simulated environment such as textures, lighting conditions, and random pedestrian behavior. To explore if the end-to-end driving overcomes this gap, the network was further trained and tested for a real environment. The training process was similar to the one defined for simulation. The driverless vehicle shown in figure 1 was driven several times in the campus with a joystick. The sensor configuration was similar to the simulation.

In real tests, it was not possible to have a ground truth and compare the testing results. Network performance is reckoned from visual inspection. The network performed well except for places with strong shadows (similar problem to simulation). Also, it drove precisely well in an unseen environment with different textures and surroundings. A sample image of the seen and unseen test area is shown in figure 13 and figure 14, respectively.

It is impossible to alter weather conditions or define pedestrian behavior in real world. The main focus of driving in a real environment was on the trained path with people obstructing the view and driving in an unseen environment.

6 CONCLUSIONS

This paper identifies factors affecting end-to-end driving for pedestrian zones. Initially, the work is done in the simulation, later it is transferred to a real system. A CNN network is designed to provide steering angles of a vehicle using RGB images from a camera mounted on the roof of a minibus. The system is tested in simulation with different weather conditions and pedestrian locations. From the results, it can be seen that the end-to-end system predicts well in the driven path with different classes of weather. If trained well for a particular environment it shows propitious results, but relying alone on this system for driving the vehicle is still not proposed; it is not known when the system goes into a failure state. Overall, the reason behind the failure was strong shadows. Also, the presence of a crowd made the vehicle slightly steer. In future work, it is planned to include depth images with extra output for handling the shadows and intersections, respectively.

REFERENCES

Argall, B. D., Chernova, S., Veloso, M., and Browning, B. (2009). A survey of robot learning from demonstration. *Robotics and autonomous systems*, 57(5):469–483.

Bojarski, M., Del Testa, D., Dworakowski, D., Firner, B., Flepp, B., Goyal, P., Jackel, L. D., Monfort, M., Muller, U., Zhang, J., et al. (2016). End to end learning for self-driving cars. *arXiv preprint arXiv:1604.07316*.

Chang, T.-H., Wang, L.-S., and Chang, F.-R. (2009). A solution to the ill-conditioned gps positioning problem in an urban environment. *IEEE Transactions on Intelligent Transportation Systems*, 10(1):135–145.

Chen, L.-C., Zhu, Y., Papandreou, G., Schroff, F., and Adam, H. (2018). Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 801–818.

He, K., Gkioxari, G., Dollár, P., and Girshick, R. (2017). Mask r-cnn. proceedings of the ieee international conference on computer vision. *URL: http://openaccess.thecvf. com/content_ICCV_2017/papers/He_Mask_ R-CNN_ICCV_2017_paper. pdf*.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Hubschneider, C., Bauer, A., Doll, J., Weber, M., Klemm, S., Kuhnt, F., and Zöllner, J. M. (2017). Integrating end-to-end learned steering into probabilistic autonomous driving. In *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, pages 1–7. IEEE.

Jan, Q. H. and Berns, K. (2021). Safety-configuration of autonomous bus in pedestrian zone. In *VEHITS*, pages 698–705.

Jan, Q. H., Kleen, J. M. A., and Berns, K. (2020a). Self-aware pedestrians modeling for testing autonomous vehicles in simulation. In *VEHITS*, pages 577–584.

Jan, Q. H., Kleen, J. M. A., and Berns, K. (2021). Simulated pedestrian modelling for reliable testing of autonomous vehicle in pedestrian zones. In *Smart Cities, Green Technologies, and Intelligent Transport Systems: 9th International Conference, SMARTGREENS 2020, and 6th International Conference, VEHITS 2020, Prague, Czech Republic, May 2-4, 2020, Revised Selected Papers 9*, pages 290–307. Springer.

Jan, Q. H., Klein, S., and Berns, K. (2020b). Safe and efficient navigation of an autonomous shuttle in a pedestrian zone. In *Advances in Service and Industrial Robotics: Proceedings of the 28th International Conference on Robotics in Alpe-Adria-Danube Region (RAAD 2019) 28*, pages 267–274. Springer.

Lee, M.-j. and Ha, Y.-g. (2020). Autonomous driving control using end-to-end deep learning. In *2020 IEEE International Conference on Big Data and Smart Computing (BigComp)*, pages 470–473. IEEE.

Levinson, J., Askeland, J., Becker, J., Dolson, J., Held, D., Kammel, S., Kolter, J. Z., Langer, D., Pink, O., Pratt, V., et al. (2011). Towards fully autonomous driving: Systems and algorithms. In *2011 IEEE intelligent vehicles symposium (IV)*, pages 163–168. IEEE.

Manaswi, N. (2018). Understanding and working with keras deep learning with applications using python. apress, berkeley.

Pomerleau, D. A. (1988). Alvinn: An autonomous land vehicle in a neural network. *Advances in neural information processing systems*, 1.

Sutton, R. S., Barto, A. G., et al. (1998). *Introduction to reinforcement learning*, volume 135. MIT press Cambridge.

Wolf, P., Groll, T., Hemer, S., and Berns, K. (2020). Evolution of robotic simulators: Using ue 4 to enable real-world quality testing of complex autonomous robots in unstructured environments. In *SIMULTECH*, pages 271–278.

Yurtsever, E., Lambert, J., Carballo, A., and Takeda, K. (2020). A survey of autonomous driving: Common practices and emerging technologies. *IEEE access*, 8:58443–58469.