DOI: 10.2195/lj_proc_koehne_en_202410_01

Leveraging human expert knowledge to automate forklift truck driving

Nutzung des menschlichen Expertenwissens zur Automatisierung des Gabelstaplerfahrens

Phil Köhne [1]
Mirko Schaper [2]
Justus Lübbehusen[2]
Benjamin Küster [1]
Malte Stonis[1]
Ludger Overmeyer [2]

IPH - Institut für Integrierte Produktion Hannover gGmbH [1]
Leibniz Universität Hannover, Institut für Transport- und Automatisierungstechnik [2]

This work explores the challenges of fully automating in-house goods transport in environments where industrial trucks like forklift trucks remain necessary due to undefined load carrier positions and shapes. Imitation Learning (IL) is identified as a promising solution for vehicle control in repetitive tasks, yet its application in intralogistics is challenging by the dynamic complexity of industrial trucks and the large dimensional space involved. A Robot Operating System 2 (ROS2) framework is introduced, enabling the acquisition of driving data from both simulation environments and real-world demonstrators. The study also presents a network architecture combining a Convolutional Neural Network (CNN) with a Long Short-Term Memory (LSTM) network, facilitating end-to-end learning from spatial and temporal image data. The framework's effectiveness is evaluated using a dataset of expert driving maneuvers to assess the generalization potential of the IL-trained network in vehicle control in different scenarios. The research aims to demonstrate the utility of the proposed framework for data acquisition and validate IL as a control approach for industrial trucks that require generalization.

[Imitation Learning (IL), industrial truck automation, intralogistics, ROS2, load handling]

In dieser Arbeit werden die Herausforderungen der Vollautomatisierung des innerbetrieblichen Warentransports in Umgebungen untersucht, in denen manuell geführte Flurförderzeuge aufgrund von undefinierten Positionen und Formen der Ladungsträger weiterhin notwendig sind. Imitation Learning (IL) wird als eine vielversprechende Lösung für die Fahrzeugsteuerung bei sich wiederholenden Aufgaben identifiziert, jedoch wird seine Anwendung in der Intralogistik durch die Komplexität der Dynamik von Flurförderzeugen und dem großen abzubildenden Dimensionsraum erschwert. Es wird ein Robot Operating System 2 (ROS2) Framework vorgestellt, dass die Erfassung von Experten Fahrdaten sowohl aus Simulationsumgebungen als auch von realen Demonstrator Fahrzeugen ermöglicht. Darüber hinaus wird eine Netzwerkarchitektur präsentiert, die ein Convolutional Neural Network (CNN) mit einem nachgeschalteten Long Short-Term Memory (LSTM) Netzwerk kombiniert, um aus Bild- und Geschwindigkeitsdaten sowohl räumliche als auch zeitliche Informationen zu erlernen. Evaluiert wird die Effektivität des Frameworks anhand eines Datensatzes mit Expertenfahrmanövern, wobei das Generalisierungspotential des trainierten Netzes für die Fahrzeugsteuerung bewertet wird. Ziel der Arbeit ist es, den Nutzen des vorgeschlagenen Frameworks für die Datenerfassung zu demonstrieren und IL als Steuerungsansatz für Flurförderzeuge zu validieren.

[Imitationslernen (IL), Flurförderzeug-Automatisierung, Intralogistik, ROS2, Lasthandhabung]

1 INTRODUCTION

The realization of fully automated, in-house goods transport is still limited to clearly defined routes and storage areas. In storage areas where no defined position or shape of the individual load carriers can be specified due to their scope of use, manually guided industrial trucks continue to be used even though the transport task of picking up a load carrier is also clearly defined in these areas. A precise and robust approach for the vehicle control is required in order to ensure the continuous and reliable transport of goods by previously manually guided industrial trucks.

In recent years, IL has shown promising results in overcoming challenges for which there is a defined solution path using expert data. IL algorithms can accurately replicate how experts behave in known environments using an end-to-end approach [1]. Thereby the accuracy and robustness of the various IL algorithms depend significantly on the amount and quality of the expert data. as shown in [2]: “if the expert and learner share the same policy space, then the policy is always imitable”.

Since industrial trucks have a high level of dynamic complexity and the dimensional space to be mapped is very large, the use of IL in intralogistics is associated with major challenges. In this context, not only a control approach with a certain degree of generalization is required, but also a variable design of the data acquisition process is necessary to be able to efficiently record data for different driving scenarios.

This thesis presents a Robot Operating System 2 (ROS2) framework that can be used for the acquisition of driving data both from the Nvidia Omniverse Isaac Sim simulation environment and from real demonstrators. Furthermore, the trained network architecture by IL is presented. It consists of a Convolutional Neural Network (CNN) in combination with a downstream Long Short-Term Memory (LSTM) to apply end-to-end learning not only to the spatial analysis of a single image, but also to the temporal information of time-sequential image frames. The data set used for training comprises various positions and orientations of a DIN EN 13698-1 Euro pallet and is recorded according to the proposed framework. Based on the used network architecture and data set, the generalization potential of the trained network architecture with IL is evaluated in known and unknown situations for the vehicle control of an industrial truck. The aim is both to present a framework for manual data acquisition in a simulated intralogistics environment utilizing a human expert driver and demonstrate IL as a control approach for industrial trucks that require a certain degree of generalization.

For this purpose, the thesis is structured as follows: The second section presents an overview of related work, while the third section presents the proposed framework. The fourth section shows the experimental setup with the results of the generalization study, and the last section summarizes the results of the work.

2 Related Work

The concept of end-to-end learning was first presented by Pomerleau et al. in 1989. They used a three-layered neural network to observe the curvature of the road, in order to output the steering angle of a land vehicle and navigate on the road [3]. This lane keeping behavior was extended by Müller et al. to include obstacle avoidance, in which depth information can be extracted via two front cameras in order to drive a small vehicle in an end-to-end approach [4]. With the continuous increase in computing power, deep neural networks have been increasingly used for the autonomous control of vehicles. Bojarski et al. extended the two-camera system and trained a 3-camera CNN-based model (DAVE-2) for steering control of a vehicle in a series of real-world driving scenarios [5]. This was one of the key points that brought end-to-end systems to the forefront of autonomous vehicle research. Another publication by Bojarski et al. focuses on identifying the objects and regions that are of crucial relevance for the network in Paper [5] to determine steering angles. The results demonstrate that the features used by the network in relation to the scene are very similar to those of a human [6]. The works in [7], [8], [9] are further examples of CNN-based end-to-end learning approaches to control the steering of an autonomous vehicle. In order to also exploit temporal dependencies between consecutive images, M. Lee and Y. Ha present a combination of CNN with an LSTM for steering angle control [10]. The combination of CNN and LSTM enabled the extraction of both temporal and spatial features. The proposed method was trained only for the steering wheel angle using a driving simulator. Hecker et al. argue that human drivers usually have access to more data than autonomous vehicles. They propose to provide end-to-end systems with input information about their entire environment. A system of eight cameras is used together with a route planner to provide high-level action information. Each camera is fed into a network consisting of several CNN and LSTM sub-networks. Subsequently, the information from the cameras and the map is fused to produce future speed and guidance predictions. The inclusion of additional information leads to a significant increase in performance [11]. Haavaldsen et al. show that by combining CNN and LSTM, end-to-end systems can operate autonomously in simple urban environments by controlling steering angle and speed. Furthermore, it is found that utilizing temporal information in subsequent images improves the system's ability to judge movement and distance [12]. Chi and Mu also show a model that uses the LSTM architecture. They model the steering angle as a continuous variable. The LSTM network is trained to minimize the loss between the predicted steering angles to those of an expert. The architecture of the network consists of two individual sub-networks. The first network processes the visual environment and the internal status of the vehicle. The second sub-network is the steering prediction sub-network, which is responsible for the steering output. The feature extraction subnet performs a spatio-temporal convolution to model the sequential learning problem of autonomous steering. The steering prediction network fuses multiple types of temporal information to make the steering, speed and torque predictions [13]. For performance analysis, the method is compared with competing algorithms such as AlexNet [14] and PilotNet [6], and outperforms them both in predicting steering angle. Le Mero et al. give an overview of the state of the art in imitation learning methods, their applications in the field of autonomous vehicles and discuss open challenges. The field is divided into three main approaches: Behavioural Cloning, Direct Policy Learning and Inverse Reinforcement Learning. As IL is a data-based approach, like many other deep learning paradigms, the overview also summarizes existing datasets and simulators and examines their potential applications [1]. This overview can serve as an initial starting point for researchers and provide a comprehensive overview of existing research.

3 Proposed Framework

3.1 Architecture and Dataset recording

To record datasets of the driving behavior of an expert forklift truck driver we are presenting a ROS2 Framework which can be applied both in real world and simulated environments. With focus on simulated environments, we are explaining the core components of our framework including a dataset recorder as well as a suited data loader to handle the recorded datasets for training a neural network.

The process of recording a dataset in a simulated environment is illustrated in Figure 1. A human expert driver is controlling the simulated forklift truck with the help of a physical gamepad. The framework utilizes the ROS2 Joy Package which takes care of reading the actual gamepad raw state. The raw input data is then forwarded to the gamepad node over a ROS2 topic. Inside the gamepad node the raw input data is translated into application specific driving commands. To mimic the real-world forklift truck driving behavior, a dedicated controller node is necessary to provide the correct joint commands for driving the forklift truck inside the simulation. This configuration depends on the simulated vehicle and must be adapted individually.

The provided sensor data of the simulated forklift truck as well as the driving commands are then recorded by the dataset generator node and stored as a ROS Bag. A ROS Bag is a file format in ROS2 for storing ROS message and topic data. We differentiate between datasets and single runs. A dataset always consists of one or more single runs where each run is representing a trajectory from the forklift trucks initial position to the forklift truck's goal position. Every dataset itself is only representing one kind of trajectory. The initial forklift truck position and the goal forklift truck position are always the same. Only the course of the trajectory differs inside the dataset. Single runs are stored as ROS Bags and multiple single runs form a dataset. All datasets are recorded with a fixed frequency which is predefined by the gamepad node. All topics are recorded with respect to the driving commands publishing frequency. Every timestep is then representing sensor data, images and driving commands. This guarantees a lossless dataset without missing any required information for training a neural network.

At every start of a new recording session a unique dataset folder is created, and the single runs are stored inside. The completed runs are then named with ascending numbers. The recording of single runs can be started and stopped with the gamepad. Additionally, it can also be decided, if the last recorded run should be saved to the dataset or not. For simulated environments an automatic environment reset can be activated to reset the simulation after a successful recording. This sets the forklift truck as well as all other simulation members to its initial positions. This leads to an easy, fast and time efficient method to record human driving behavior.

For our experiment we use Nvidia's Omniverse Isaac Sim to generate a realistic simulation environment. The virtual forklift truck model "atlas" is taken from Nvidia's sample assets. It comes fully rigged and articulated and is simulation ready. We are controlling the simulated forklift truck with a normalized driving command in form of a linear velocity and a steering angle in the range from -1 to 1. The controller node is converting this driving command into joint specific arguments to control the back wheel speed and the back wheel swivel position of the simulated forklift truck. Inside the controller node we are also restricting the forklift trucks maximum velocity and steering angle.

The simulated forklift truck is publishing a twist message with information about its actual velocity in free space

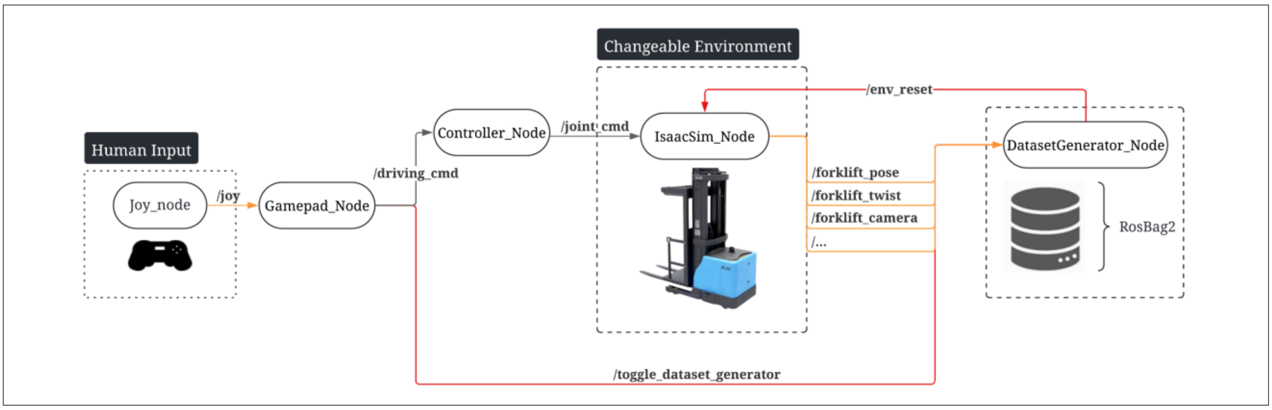


Figure 1. Overview of the ROS2 Framework for recording a Dataset.

broken into its linear and angular parts. To provide a deeper insight into the recorded trajectory, the actual forklift trucks absolute position is also recorded. The simulation environment is observed with an omniscient top view camera attached to the front of the forklift truck. This camera presents a 2D top view from a bird's eye perspective. Together with the normalized driving commands, this data is recorded into a dataset with the help of the dataset generator node and used for training a neural network.

3.2 Dataset Preprocessing

The recorded datasets will be used to train a model to imitate the expert's driving behavior. Therefore, the recorded data needs to be read in by a data loader and parsed into a suitable format for processing inside the neural network. We are using our own data loader which can parse the recoded ROS Bags in a fast and memory efficient manner. The data loader aims to address the key challenge of the slow reading speed of ROS Bags especially when training a neural network over multiple epochs. Reading in ROS Bags file by file, the preprocessing of the recorded data will be necessary every time a new training epoch is started. This is a time consuming and slow process. Therefore, we are using our own, faster data loader. The recorded ROS Bags will only be parsed once and the desired data for training the neural network will then be preprocessed. The process is explained in detail based on an image topic:

First, the recorded image topic will be read in from the ROS Bag and parsed into a suitable format. Then, the image will be resized, cropped and normalized. The preprocessed image is then stored and saved inside a temporary pickle file. Doing this for every desired topic inside the ROS Bags will result in a single pickle file which can be fed into a neural network fast.

For our Experiment we are resizing the recorded RGB images from the top view camera from 3 x 960 x 960 pixels to 3 x 160 x 160 pixels. Together with the recorded normalized driving commands and the forklift trucks twist velocities we are saving the images to the temporary pickle file.

3.3 Model Architecture

We developed our model architecture for IL inspired by [15], opting to design and implement a custom CNN for image classification instead of utilizing a pretrained ResNet model. Initially, the observed top view images undergo normalization before being processed by the CNN. Concurrently, the forklift trucks linear and angular velocities in free space are normalized and processed through a Multilayer Perceptron (MLP) network. The outputs from both networks are then concatenated and input into a two-layer LSTM network. The Model architecture is illustrated in Figure 2.

The LSTM network's purpose is to incorporate a form of memory into our model, facilitating meta-learning [16], which enhances the model's ability to generalize effectively even with limited training data. Following the LSTM network, a classification MLP is employed to project the LSTM output onto two scalar values, corresponding to the forklift truck's driving commands.

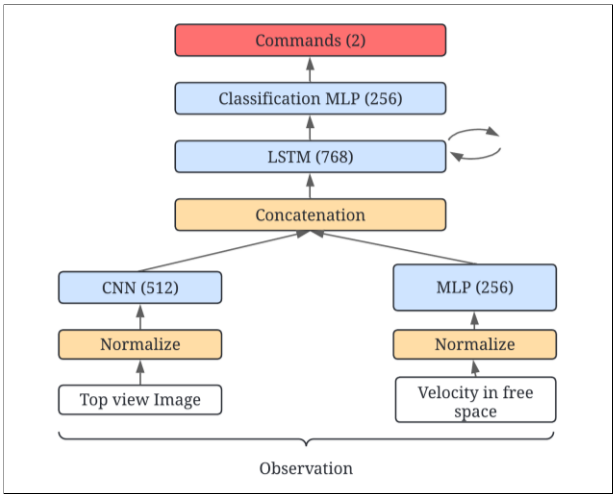


Figure 2. Our model architecture consists of a combination of CNN and LSTM networks as well as an MLP classification network inspired by [15].

For training the network weights we utilized the mean squared error (MSE) as the loss function and adopted the Adagrad optimizer with a fixed learning rate of 0.0005. This approach ensures the robustness and adaptability of our model in dynamic environments.

3.4 Deploying the Trained Model

We have trained our model for 300 epochs on a cluster of two Nvidia RTX A6000 GPUs. The trained model is deployed via its own dedicated neural network node which can be seen in Appendix 1. The trained model can generate normalized driving commands in the range from -1 to 1 and send them directly to the controller node. The image input data for the model is the omniscient 2D camera top view and the normalized linear and angular velocity of the forklift truck in free space. Both are preprocessed within the ROS node itself. If the pallet is reached correctly, the simulated environment resets to the starting position itself.

4 Experimental Setup

4.1 Experimental Design

The experiments were designed to simulate a real-world scenario where the framework's capabilities in recording a training dataset and controlling a forklift truck with a trained model can be reviewed. The focus lays on the evaluation of the capabilities of the trained model when providing the top view image. It will be critically assessed how the model performs and how the given task is accomplished. Therefore, we are describing a scenario where an expert driver is told to pick up a pallet in different rotations and positions. The palette is rotated and positioned in the following ways:

- In the first situation, the longitudinal axes of the forklift truck and pallet are aligned with a 10-meter gap between them. In the second and third situation, the forklift remains in its initial position while the pallet is shifted 5 meters to the left and right, respectively.

- The pallet is initially aligned with the forklift truck at a 0-degree angle. The angle is then adjusted in increments of 5 degrees, ranging from -45 to +45 degrees

The different Positions are exemplary shown in Figure 3. This leads to 57 possible situations. The expert human driver is told to drive each situation five times inside the simulated environment. Each run starts at the initial start position and ends when the forks of the forklift truck are fully inserted into the pallet. Taking this into account, the complete dataset consists out of 285 recorded single runs.

The environment inside the simulation consists of a flat grey surface where the forklift truck and the pallet are placed on. From above, a simulated light will illuminate the area and ensures constant lighting conditions. There are no other objects within the simulated environment that could influence or distract the neural network. When recording the training data set, it is ensured that the pallet is always fully recognizable in the top view RGB image. The simulated forklift truck is restricted to a maximum linear speed of 1.6 m/s, while the back wheel swivel is limited to a maximum steering angle of ±90 degrees.

4.2 Training Dataset diversity

A high level of diversity in the training data is a fundamental requirement and significantly influences the performance of the trained model [17]. A dataset that is insufficiently diverse can readily result in overfitting and poor generalizability [18]. To address this issue, it was of importance to ensure comprehensive coverage of potential pallet orientations and positions during data collection. Figure 3 illustrates exemplary the recorded trajectories for three pallet positions and three different pallet rotations derived from the training data set. The blue lines indicate trajectories to pallets with a relative angle to the forklift truck of 0 degrees. The red lines represent trajectories to pallets with a relative angle of -45 degrees. Respectively, trajectories in green indicate a relative rotation of the pallet of 45 degrees. As illustrated, the trajectories entered by the expert human driver and stored in the data sets exhibit slight dispersion, ensuring that no trajectory occurs more than once in the training data set. Each potential pallet rotation and position is represented by precisely five distinct trajectories, thereby reducing bias by ensuring that no situation is under- or over-represented in the training data set.

The distribution of driving data within the entire datasets, derived from the entered trajectories, can be observed in Figure 4 and Figure 5. The percentages of the normalized drive command ranges in the total number of drive commands are shown. For the linear velocity drive command, the range from 0 to 1 is divided into five equally sized areas. Similarly, the steering drive command range from -1 to 1 is divided into eight areas.

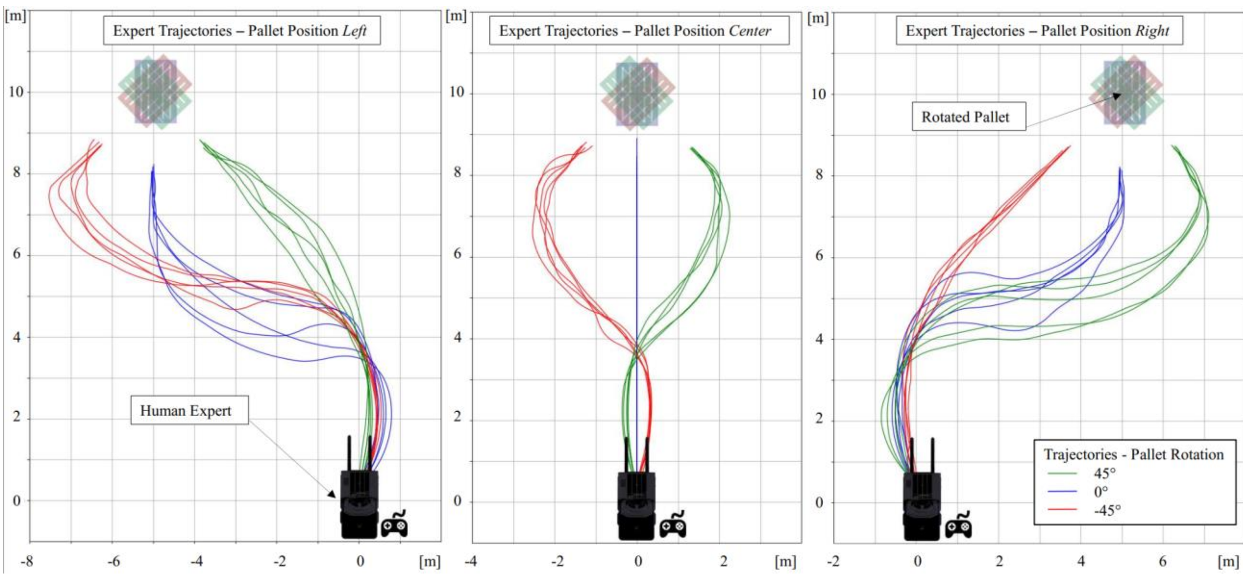


Figure 3. Exemplary trajectories of the recoded Dataset by a human expert driver. Blue lines indicate a Pallet orientation of 0 degrees. Red trajectories demonstrate a pallet rotation of 45 degrees and green lines indicate trajectories for a rotation of -45 degrees

As illustrated in Figure 4, the majority of driving occurred within the medium speed range represented by the large amount of linear velocity commands from 0.4 to 0.7. Linear velocity commands that are both very slow and very high represent a significantly smaller proportion of the total. The reason for this is that the expert adjusts the driving behavior depending on the distance between the forklift truck and the pallet. This natural behavior introduces a distortion to the linear velocity driving commands within the data set.

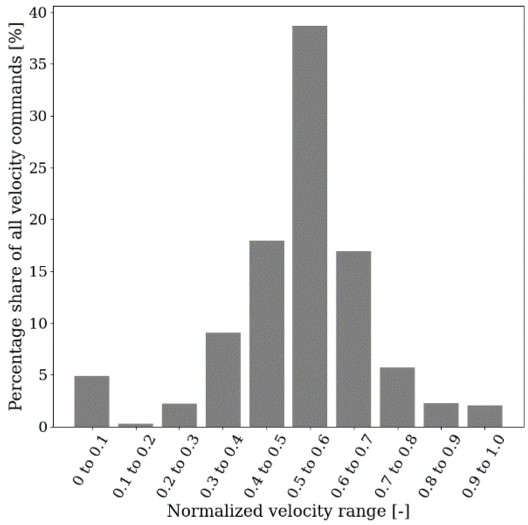
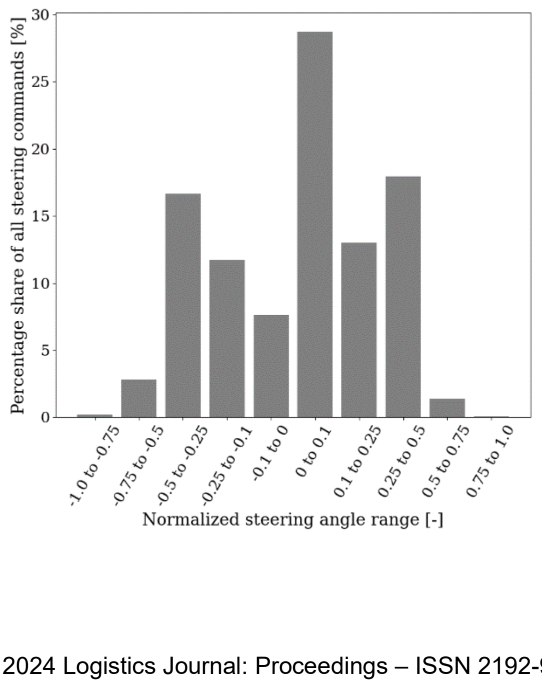


Figure 4. Distribution of the normalized velocity commands in the recorded training dataset

Figure 5. Distribution of the normalized steering commands in the training dataset

Figure 5 illustrates the distribution of the steering commands. It can be seen that very strong steering commands do not make up a significant proportion of the data set. The data set is dominated by steering commands in the medium range, which are used for normal lane corrections. The distribution of steering commands between the positive and negative range is symmetrical. Only in the range from 0 to 0.1 a bias towards the positive range can be observed. This slight bias in the range of 0 to 0.1 is possibly caused by the background noise of the gamepad. When looking at Figure 3 this slight distortion is not visible. The symmetrical arrangement of the palettes also results in symmetrical expert trajectories, suggesting that the distortion caused by gamepad input has no significant effect.



4.3 Evaluation and Results

To examine the trained model, we follow two different strategies. First, we present the model with known situations that are completely contained in the training data set. The aim is to assess whether the model is able to reproduce human expert knowledge. In the second part of the study, we present the model with new unknown situations that are not included in the training data set. The situations presented are similar to those in the training dataset and represent only slight variations. The aim of this study is to evaluate the generalization ability of the trained model and the network architecture used. For both investigations, we use the framework from Appendix 1. The simulated forklift truck controlled by the trained model is referred to as the agent in the following.

Just like the human expert, the agent must also complete the situations described in section 4.1 in our study. To do this, the agent steers the forklift truck to the pallet based on the omniscient top view camera and the forklift trucks velocity data. For each situation, the agent is given five attempts to solve it successfully. A trip is considered successful if the forks of the forklift truck are fully immersed in the pallet from the correct side and the forklift truck stops automatically. If the forklift truck touches or moves the pallet, this is not taken into account in the assessment.

4.3.1 Investigation of the Reproducibility of the Expert Trajectory

The results for the first part of the study are shown in the table of Appendix 2. The percentage of successfully completed trips is shown for the different positions and orientations of the pallet. The color gradient from green (all five trips successfully completed) to red (no trip successfully completed) highlights the result once again. It is easy to see that a pallet that is not subject to a lateral position shift can be approached most successfully by the agent. This is also shown graphically in the center of Figure 6, with the expert trajectories highlighted transparently. The deviation of the selected trajectory of the agent is small compared to that of the expert. If the pallet is oriented by -45 degrees (green trajectories), the curve is less pronounced. An orientation of the pallet around 0 degrees (blue trajectory) was approached by the expert without any steering movements. The agent also successfully approaches this pallet orientation but achieves this with a slightly curved trajectory.

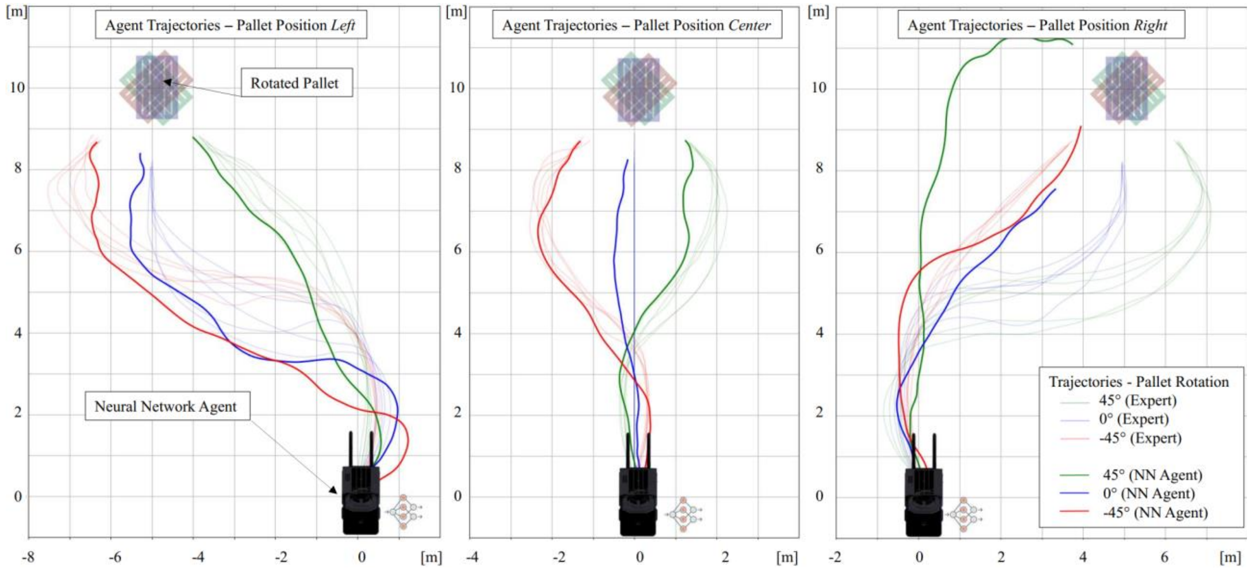


Figure 6. Exemplary trajectories of the trained neural network agent (thick lines) in contrast to the experts' trajectories (transparent lines) for known situations.

A pallet oriented with a negative relative angle is approached more poorly by the agent. This can also be observed if the position of the pallet is shifted in a negative direction. Figure 6 on the left also shows that positively oriented pallets are approached by the agent without any problems. In this case, the specified expert trajectory is reproduced to a lesser extent, but the pallet is still approached successfully. The agent also successfully approaches pallets with an orientation of 0 degrees on trajectories very similar to those of the expert. Moving the pallet in a positive direction poses major problems for the agent. None of the pallet orientations presented could be approached satisfactorily. This can also be seen in Figure 6 on the right, where the agent attempts to approach the pallet from the opposite side when it is rotated by -45 degrees. The deviation from the expert trajectory in the training data set is significant in this case. Only the pallet oriented at 45 degrees was approached by an expert-like trajectory. The pallet oriented at 0 degrees could not be approached by the agent.

In principle, the knowledge contained in the training data set can be reproduced by the agent. Extreme positive or negative angles in the orientation pose a challenge for the agent. Approaching positions that are shifted in a positive direction cannot be solved satisfactorily. One possible reason for this behavior is the bias in the training data, which contains slightly more steering commands in one direction. Moreover, the number of expert trajectories utilized for agent training may be insufficient.

4.3.2 Investigation of generalizability in unknown situations

The generalization capability of the agent is examined with the help of a new, unknown position and with new, unknown rotations of the pallet. Based on the results from the reproducibility study, it was decided to examine only a negative shift of -2.5 m of the pallet. The results can be seen in the table of Appendix 3 for the unknown position with known rotation and in Appendix 4 for the unknown position and unknown rotation.

A slightly altered pallet position with small rotational deviations does not pose a challenge, and the pallet is approached correctly. The pallet is approached correctly. Larger angle changes in the rotation of the pallet cannot always be approached correctly. Unlike in the first part of the experiment, pallets rotated in a positive direction are more difficult situations to solve. This can also be seen in Figure 7 on the left. To rotate the pallet by 45 degrees, for example, the agent must make a double turn.

The pallet, whose position and rotation are unknown, shows very similar results when approached by the agent. A slight deviation in the rotation of the pallet does not mean that it can no longer be approached. Only with larger rotations in the positive range do difficulties arise again when the agent approaches the pallet.

Overall, the agent solves the unknown situations well. The trained model is able to derive driving methods for new, unknown situations from previously learned knowledge. A change in the angle of rotation of the pallet is tolerated more readily than a change in position. The reason for this is that a pallet that has only changed its angle requires significantly less adaptation in the driving maneuver than a pallet that has changed its position. Furthermore, the data set contains significantly more situations with differently rotated pallets than pallets in different positions. The data set contains three different pallet positions. On the other hand, there are 19 different pallet orientations.

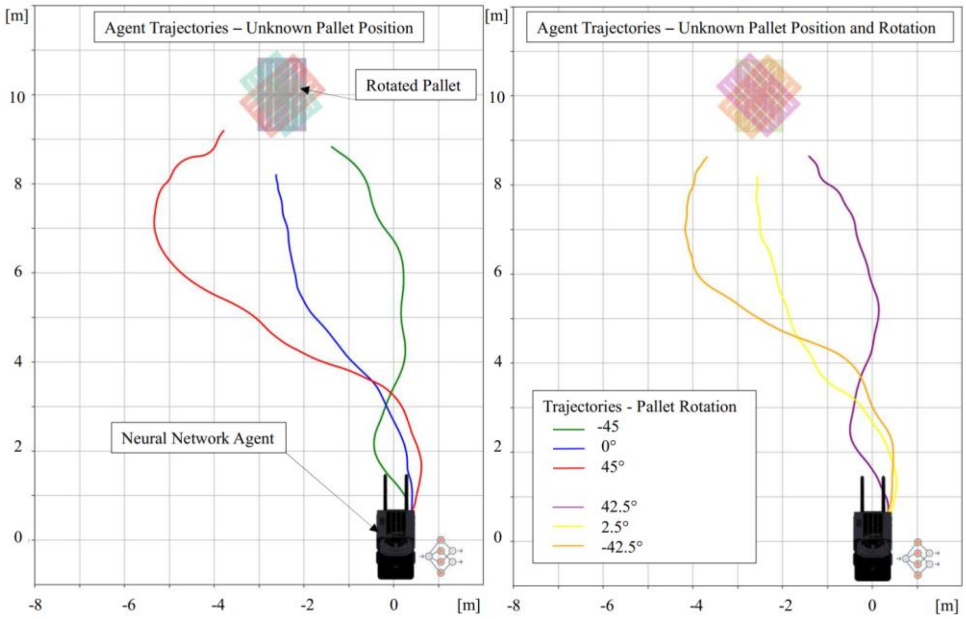


Figure 7. Exemplary trajectories of the trained neural network for an unknown pallet position (left) and unknown pallet position and rotation (right).

5 CONCLUSION

In this paper we present a ROS2 framework that can be used to record driving data sets in the simulation environment Nvidia Omniverse Isaac Sim as well as on a real demonstrator. With the help of the presented framework, expert trajectories of a simulated forklift truck are recorded within the simulation environment. The aim was to successfully pick up pallets at different positions and with different orientations relative to the forklift truck. The driving data set includes information about the executed driving command, a top view 2D camera image from a bird's eye view and the current vehicle speed. The recorded human expert trajectories form a training data set which is then used to train a suitable neural network architecture. By training with the recorded expert data, the trained model is enabled to independently send driving commands to the simulated forklift truck based on the environment information. The performance of the trained model is evaluated using two different strategies. Situations contained in the expert data set are used to evaluate the extent to which the expert knowledge can be reproduced by the trained model. The generalization ability of the model is evaluated by showing unknown situations (situations that are not contained in the training data set). It was found that it is possible to reproduce the expert knowledge. Difficulties are encountered when approaching the pallet, particularly when the relative angle to the forklift truck is very large. Good generalization was also observed in situations unknown to the model. Moving the pallet to an unknown position posed a greater challenge for the trained model than turning a pallet to an unknown angle. It could be shown that even with a limited amount of expert data, the expert knowledge can be reproduced well by a trained model. Furthermore, the trained model was able to understand slight changes in the position and rotation of the pallet and adapt its driving style accordingly. The ROS2 framework presented here is therefore suitable for collecting data of the driving behavior of a human expert driver inside a simulated environment. In the Future the utilization of the presented framework will be extended to a real forklift truck. For this, the top view camera position will also be replaced by a camera position that is permissible on a real forklift truck. The situations shown here will then be solved by a trained neural network on a real forklift truck.

6 FUNDING

This Project is funded by the Federal Ministry for Economic Affairs and Climate Protection (BMWK) as part of the “New Vehicle and System Technologies” program and supervised by the project management agency TÜV Rheinland based on a resolution of the German Parliament.

LITERATURE

[1] L. Le Mero, D. Yi, M. Dianati, und A. Mouzakitis, „A Survey on Imitation Learning Techniques for End-to-End Autonomous Vehicles“, *IEEE Trans. Intell. Transport. Syst.*, Bd. 23, Nr. 9, S. 14128–14147, Sep. 2022, doi: 10.1109/TITS.2022.3144867.

[2] J. Zhang, D. Kumor, und E. Bareinboim, „Causal Imitation Learning with Unobserved Confounders“,

12. August 2022, *arXiv*: arXiv:2208.06267. Zugegriffen: 14. August 2024. [Online]. Verfügbar unter: http://arxiv.org/abs/2208.06267
[3] D. A. Pomerleau, „ALVINN: An Autonomous Land Vehicle in a Neural Network“.
[4] Y. LeCun, U. Muller, J. Ben, und E. Cosatto, „Off-Road Obstacle Avoidance through End-to-End Learning“.
[5] M. Bojarski *u. a.*, „End to End Learning for Self-Driving Cars“, 25. April 2016, *arXiv*: arXiv:1604.07316. Zugegriffen: 27. Mai 2024. [Online]. Verfügbar unter: http://arxiv.org/abs/1604.07316
[6] M. Bojarski *u. a.*, „Explaining How a Deep Neural Network Trained with End-to-End Learning Steers a Car“, 25. April 2017, *arXiv*: arXiv:1704.07911. Zugegriffen: 8. August 2024. [Online]. Verfügbar unter: http://arxiv.org/abs/1704.07911
[7] V. Rausch, A. Hansen, E. Solowjow, C. Liu, E. Kreuzer, und J. K. Hedrick, „Learning a deep neural net policy for end-to-end control of autonomous vehicles“, in *2017 American Control Conference (ACC)*, Seattle, WA, USA: IEEE, Mai 2017, S. 4914–4919. doi: 10.23919/ACC.2017.7963716.
[8] Z. Chen und X. Huang, „End-to-end learning for lane keeping of self-driving cars“, in *2017 IEEE Intelligent Vehicles Symposium (IV)*, Los Angeles, CA, USA: IEEE, Juni 2017, S. 1856–1860. doi: 10.1109/IVS.2017.7995975.
[9] M. G. Bechtel, E. Mcellhiney, M. Kim, und H. Yun, „DeepPicar: A Low-Cost Deep Neural Network-Based Autonomous Car“, in *2018 IEEE 24th International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA)*, Hakodate: IEEE, Aug. 2018, S. 11–21. doi: 10.1109/RTCSA.2018.00011.
[10] M. Lee und Y. Ha, „Autonomous Driving Control Using End-to-End Deep Learning“, in *2020 IEEE International Conference on Big Data and Smart Computing (BigComp)*, Busan, Korea (South): IEEE, Feb. 2020, S. 470–473. doi: 10.1109/BigComp48618.2020.00-23.
[11] S. Hecker, D. Dai, und L. Van Gool, „End-to-End Learning of Driving Models with Surround-View Cameras and Route Planners“, 6. August 2018, *arXiv*: arXiv:1803.10158. Zugegriffen: 8. August 2024. [Online]. Verfügbar unter: http://arxiv.org/abs/1803.10158
[12] H. Haavaldsen, M. Aasboe, und F. Lindseth, „Autonomous Vehicle Control: End-to-end Learning in Simulated Urban Environments“, 16. Mai 2019, *arXiv*: arXiv:1905.06712. Zugegriffen: 8. August 2024. [Online]. Verfügbar unter: http://arxiv.org/abs/1905.06712
[13] L. Chi und Y. Mu, „Deep Steering: Learning End-to-End Driving Model from Spatial and Temporal Visual Cues“, 12. August 2017, *arXiv*: arXiv:1708.03798. Zugegriffen: 8. August 2024. [Online]. Verfügbar unter: http://arxiv.org/abs/1708.03798
[14] A. Krizhevsky, I. Sutskever, und G. E. Hinton, „ImageNet classification with deep convolutional neural networks“, *Commun. ACM*, Bd. 60, Nr. 6, S. 84–90, Mai 2017, doi: 10.1145/3065386.
[15] OpenAI *u. a.*, „Learning Dexterous In-Hand Manipulation“, 18. Januar 2019, *arXiv*: arXiv:1808.00177. Zugegriffen: 8. August 2024. [Online]. Verfügbar unter: http://arxiv.org/abs/1808.00177
[16] M. Huisman, T. M. Moerland, A. Plaat, und J. N. Van Rijn, „Are LSTMs good few-shot learners?“, *Mach Learn*, Bd. 112, Nr. 11, S. 4635–4662, Nov. 2023, doi: 10.1007/s10994-023-06394-x.
[17] H. Sug, „Performance of Machine Learning Algorithms and Diversity in Data“, *MATEC Web Conf.*, Bd. 210, S. 04019, 2018, doi: 10.1051/matecconf/201821004019.
[18] Z. Gong, P. Zhong, und W. Hu, „Diversity in Machine Learning“, *IEEE Access*, Bd. 7, S. 64323–64350, 2019, doi: 10.1109/ACCESS.2019.2917620.

Phil Köhne, M.Eng., Project Engineer, IPH – Institut für Integrierte Produktion Hannover gGmbH.

Phil Köhne was born 1997 in Hannover, Germany. Between 2016 and 2023 he studied Mechanical Engineering at the University of applied sciences and arts Hannover. Since 2023, he has been working at IPH in the Production Automation department with focus on autonomous driving technologies for use in intralogistics and robotics applications.

Address: IPH – Institut für Integrierte Produktion Hannover gGmbH, Hollerithalle 6, 30419 Hannover, Germany, Phone: +49 511 27976-233, Fax: +49 511 27976-888, E-Mail: koehne@iph-hannover.de

Mirko Schaper, M. Sc., Research Associate, ITA - Institute of Transport and Automation Technology

Mirko Schaper, born in 1995 in Langenhagen, Germany, studied mechanical engineering at Leibniz Universität Hannover with a specialisation in development and design of mechatronic systems. While pursuing his Master's degree, he worked as a part-time software developer at Dream Chip Technologies GmbH. Since completing his M.Sc. in 2023, he has been working in the automation working group at ITA, concentrating on autonomous driving in intralogistics and robotics.

Address: Institute of Transport and Automation Technology, Leibniz Universität Hannover, An der Universität 2, 30823, Garbsen, Germany,
Phone: +49 511 762-18286, Fax: +49 511 762-4007, E-Mail: mirko.schaper@ita.uni-hannover.de

Justus Lübbehusen, M. Sc., Research Associate, ITA – Institute of Transport and Automation Technology

Justus Lübbehusen, born in 1998 in Hannover, Germany, pursued his studies in Engineering and Business Administration at Leibniz Universität Hannover, with a focus on Digitalization and Automation, from 2018 to 2024. Upon earning his M.Sc. in 2024, he joined the transport group at ITA, where he specializes in object detection and pose estimation within intralogistics and robotics.

Address: Institute of Transport and Automation Technology, Leibniz Universität Hannover, An der Universität 2, 30823 Garbsen, Germany,

Phone: +49 511 762-18339, Fax: +49 511 762-4007, Email: justus.luebbehusen@ita.uni-hannover.de

Dr.-Ing. Benjamin Küster., Head of Production Automation Department, IPH – Institut für Integrierte Produktion Hannover gGmbH

Dr.-Ing. Benjamin Küster (*1988) studied industrial engineering at Leibniz Universität Hannover, specializing in production engineering and business informatics. He has been working as a project engineer in the field of production automation since November 2014. He has been Head of Production Automation since September 2017.

Dr.-Ing. Malte Stonis, Coordinating Managing Director, IPH – Institut für Integrierte Produktion Hannover gGmbH

Dr.-Ing. Malte Stonis (*1979) studied mechanical engineering at Leibniz Universität Hannover, specializing in vehicle systems and biomedical engineering. He has worked at IPH since 2006, initially as a project engineer in the Process Technology department and from 2008 as head of department. Malte Stonis has been Coordinating Managing Director of IPH since September 2016.

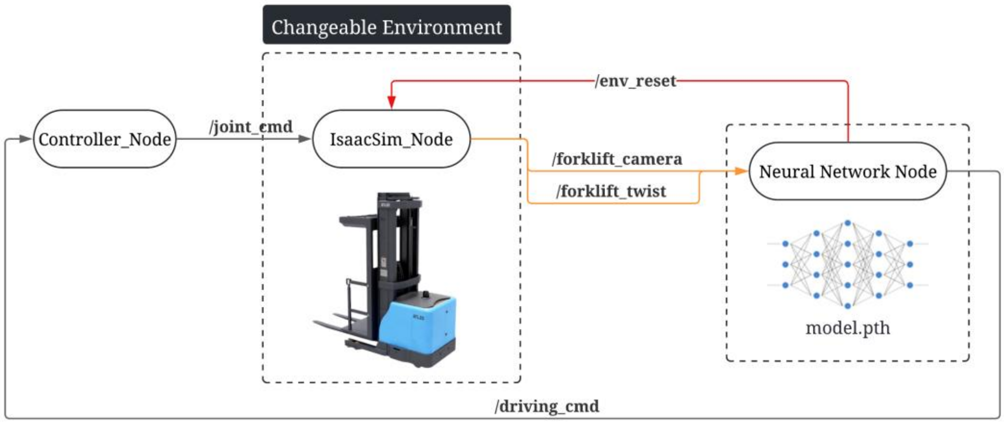
Prof. Dr.-Ing. Ludger Overmeyer, Managing Director, IPH – Institut für Integrierte Produktion Hannover gGmbH, Head of Institute of Transport and Automation Technology (ITA), Leibniz University Hannover

Prof. Dr.-Ing. Ludger Overmeyer, born in 1964, studied electrical engineering at the University of Hanover and then worked as a research assistant at the Laser Zentrum Hannover e.V. (LZH). After obtaining his doctorate in engineering, he worked as a manager in the field of research and development in mechanical and plant engineering for the semiconductor industry. Since 2002, he has headed the ITA at Leibniz University Hannover. In 2007, he took on the role of managing partner of IPH.

Address: Institute of Transport and Automation Technology, Leibniz Universität Hannover, An der Universität 2, 30823 Garbsen, Germany

DOI: 10.2195/lj_proc_koehne_en_202410_01

Appendix 1. Deploying the trained model in the ROS2 Framework.



Appendix 2. Percentage of successfully accomplished trips for known Situations by the trained neural network agent.

Pallet position		Pallet orientation																		
x	y	-45	-40	-35	-30	-25	-20	-15	-10	-5	0	5	10	15	20	25	30	35	40	45
0	-10	0,2	0,6	0,8	0,8	0,4	0,2	0,8	0,4	0,6	1	0,8	0,6	0,6	0,8	1	0,6	1	0,6	0,8
-5	-10	0,6	0,8	0,4	0,8	0,6	0,2	1	0,8	0,6	0,8	1	1	0,8	1	0,8	0,8	0	1	1
5	-10	0	0	0	0,2	0,2	0,2	0,2	0,2	0,2	0	0	0,4	0,4	0,2	0,2	0,6	0,2	0,2	0,2

Appendix 3. Percentage of successfully accomplished trips for unknown positions and known orientations of the pallet by the trained neural network agent.

Pallet position		Pallet orientation																		
x	y	-45	-40	-35	-30	-25	-20	-15	-10	-5	0	5	10	15	20	25	30	35	40	45
-2,5	-10	0,8	1	1	1	1	0,8	0,4	1	1	1	1	1	0,8	0,4	1	1	0,6	0,6	0,6

Appendix 4. Percentage of successfully accomplished trips for unknown positions and unknown orientations of the pallet by the trained neural network agent.

Pallet position		Pallet orientation																	
x	y	-42,5	-37,5	-32,5	-27,5	-22,5	-17,5	-12,5	-7,5	-2,5	2,5	7,5	12,5	17,5	22,5	27,5	32,5	37,5	42,5
-2,5	-10	1	1	1	1	1	0,8	0,8	1	0,8	1	1	1	0,8	1	1	0,6	0,6	0,2