R-PEAK DETECTION FROM NOISY ECG DATA USING MULTI-CHANNEL 1D-CNN WITH ACCELEROMETER INPUT

Tetsuya Hirota
Ryugo Fujita
Atsushi Harada
Daisuke Kawamura
Tokai Rika Co., Ltd.
Japan

Keiichi Yamada
Meijo Univ.,
Japan

Paper Number 23-0047

ABSTRACT

In order to prevent traffic accidents due to abrupt changes in the driver's health condition, we have proposed a non-contact type electrocardiographic sensor that monitors the electrocardiogram (ECG) of a driver holding a steering wheel while seated. However, the heart rate detection accuracy degrades while driving due to the lower signal-to-noise ratio (SNR) of the ECG caused by the noise from vehicle vibration and static electricity, among others. In this study, we propose a method of detecting R-peaks of the ECG from the low SNR ECG signal with high accuracy using a multi-channel one-dimensional convolutional neural network with accelerometer signals as an input. As the results, we achieved an *F*-score of 78.5% and a root-mean-square error (RMSE) of 1.99 ms. The R-peak detection performance was significantly improved when the input data length of around 1100 ms was chosen.

1. INTRODUCTION

Today the number of car accidents still remains at a high level. It is believed that many cases of the accidents are attributable to human errors such as carelessness of the driver or violation of the Road Traffic Law; on the other hand, there are not a few cases attributable to abrupt changes in the health condition of the driver, which was caused by his/her underlying illness.

In recent times, heart diseases are the leading cause of death [1], which accounts for a large percentage of car accidents caused by driver illness [2]. Because the risk of heart diseases increases exponentially with age, considering that many countries will face an aging society and the number of elderly drivers is expected to increase in the future, the number of traffic accidents caused by heart diseases is expected to increase.

Therefore, it is one of the urgent tasks to develop a system that can detect cardiovascular abnormalities that arise during driving a vehicle by monitoring the heart activities such as the ECG or heart rate of the driver, and carry out appropriate driving interventions, such as moving the vehicle to the shoulder and safely stopping it, and take appropriate measures to rescue the driver, such as notifying the abnormality of the driver to other vehicles or calling of the emergency services.

The purpose of this study is to accurately detect the driver's heart rate interval (R to R interval, RRI) by using a non-contact method without the need to attach electrodes to the body surface. For heart rate variability (HRV) analysis [3], RRI should be acquired with high accuracy of about several milliseconds. A capacitively coupled electrocardiographic (cECG) sensor is one of the typical methods for monitoring the heartbeat in a contactless manner. However, it is difficult to obtain RRI with high enough accuracy with the method while driving, because noise caused by vehicle vibrations etc. will superimpose the cECG signals to decrease the signal-to-noise ratio (SNR) [4].

In this study, we propose a method to accurately detect the driver's RRIs in a moving vehicle by using the accelerometer signal of the vehicle together with the cECG signal. A one-dimensional convolutional neural network (1D-CNN) using cECG signals and accelerometer signals as multichannel inputs is used to detect the R-peaks for acquiring RRI.

The proposed method is evaluated by experiments using data acquired from 4 subjects while driving and the effectiveness of the method is demonstrated. The method detects R-peaks in the low SNR cECG signal with a *F*-score of 78.5% when the input window length is 1100 ms.

2. RELATED WORKS

Methods for monitoring the activity of the heart fall roughly into two categories: contact type methods and non-contact type methods.

In the contact type methods, the potential differences between two or more body surfaces sandwiching the heart are obtained by attaching the electrodes to the surfaces, but this forces the driver to attach the electrodes every time he or she gets on the vehicle, and therefore is unrealistic to apply to driver monitoring.

On the other hand, the non-contact type method does not have such disadvantages, but has a problem that the signal quality is unstable because the signal is easily affected by temperature, humidity, body movement or static electricity.

Non-contact monitoring of heart activity includes electrocardiogram monitoring using a cECG sensor [5], ballistocardiogram monitoring [6], magnetocardiography monitoring using a magnetic impedance sensor [7], and Doppler sensing [8], and several experiments of monitoring heart activity in running vehicles using these methods have been reported [4] [9] [10]. Among them, a cECG sensor is relatively resistant to noise as compared with other non-contact type sensors.

ECG waveform of one heartbeat is composed of five consecutive waves, P, Q, R, S, and T waves, as shown in Figure 1. An adaptive correlation filter [11] can be used to detect signals such as QRS signals from noisy ECG data. However, the detection fails if the noise intensity becomes greater than the signal intensity.

R-peak detection from noise-intensive ECG acquired in a running vehicle using CNN are reported [12] [13] [14], but the precision level required for HRV analysis has not been attained.

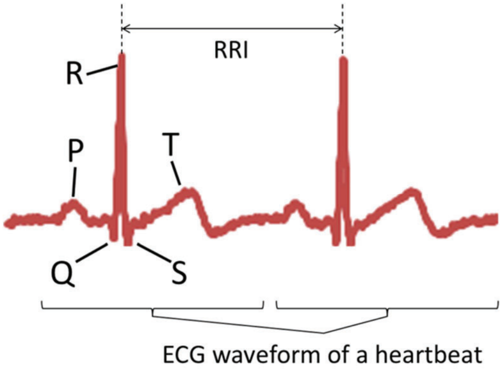


Figure 1. A typical ECG waveform. An ECG waveform of one heartbeat is composed of five consecutive waves, P, Q, R, S, and T waves.

3. PROPOSED METHOD

The architecture of the proposed method is shown in Figure 2.

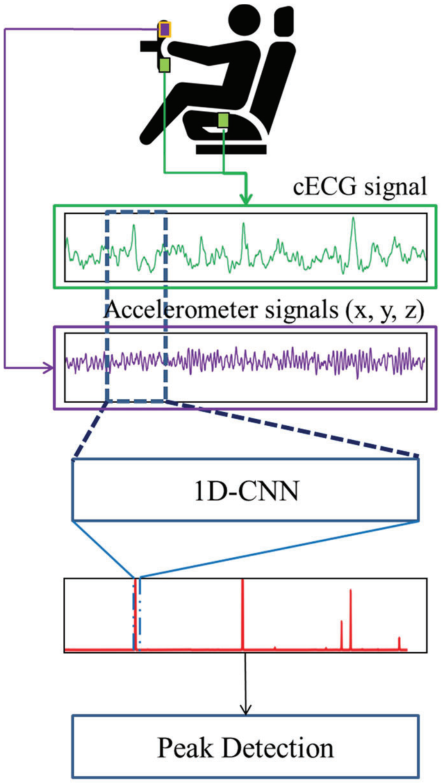


Figure 2. Schematic overview of the proposed method

We propose a 1D-CNN model for inferring the probability of the presence of the R-peak of an ECG at each moment from low SNR cECG signal and accelerometer signals acquired synchronously.

Considering that the low SNR of the cECG signal is due to the noise caused by vehicle vibrations, accelerometer signals, which are considered to be correlated with vehicle vibrations, are used as inputs of the model to remove the noise.

Given that an ECG waveform is composed of five consecutive waves, P, Q, R, S, and T waves (see Figure 1), in order to detect R-peaks, inclusion of the information of P, Q, S, and T waves within the same beat as the R-peak would be effective. Therefore, 1D-CNN layers are adopted because they are able to incorporate local time series relations of inputs.

The input to the 1D-CNN is the amplitude data of the cECG and accelerometer signals for a specific duration, and the output is the existence probability of the R-peak at the center of the duration. The R-peak timing is inferred by detecting the moment when the probability exceeds a threshold and detecting the local maximum of the probability.

4. EXPERIMENT

4-1 Data

We evaluated the proposed method with cECG signals acquired using a cECG system integrated into the passenger seat of a car [4] and acceleration signals acquired using a three-axis accelerometer attached to the steering wheel. For the reference signal, a contact-type ECG sensor (NeXus-10 Mark II, a multi-sensor physiological measurement system made by Mind Media Co.) with adhesive electrodes was used.

We acquired cECG signals and reference ECG signals of 6 subjects seated on the passenger seat of a running vehicle, and used data from 4 subjects whose reference ECG signals were measured with sufficient intensity. The total length of the data from the 4 subjects was 20 minutes.

The sampling rate was 1000 Hz, 2048 Hz and 1000 Hz for the cECG signal, reference contact-type ECG signal, and the accelerometer signal, respectively, and the contact-type ECG signal was subsequently downsampled to 1000 Hz.

4-2 Model Overview

The structure of the 1D-CNN model used in the experiment is shown in Figure 3. The input of the model was the cECG and accelerometer signals for the duration of 500 ms (500 points in total, since sampling is performed at 1000 Hz), and the output was the existence probability of the R-peak at the moment 250 ms from the beginning.

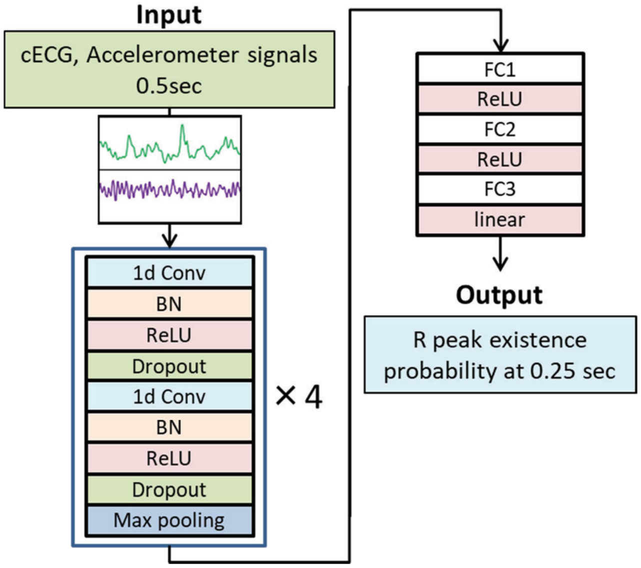


Table 1. Parameters of the network

Layer	Parameter
1dConv	kernel size:16 filters:64
Dropout	0.2
Max pooling	2
FC1	output units:16
FC2	output units:4
FC3	output units:1

Figure 3. The architecture of the network

Each convolution block consisted of two 1-dimensional convolutional layers (1d Conv), two dropout layers, two batch normalization layers (BN), two activation functions (ReLU), and one pooling layer (Max Pooling). After repeating the block four times, three fully connected layers (FC) were applied. The parameters for each layer of the network are shown in Table 1. All the layers with the same function (layer name) had the same parameter values.

4-3 Training

In training the 1D-CNN model, 500 ms-length cECG and accelerometer signal sections were extracted and used as inputs, and R-peak labels created from the reference signals were used as annotations. Specifically, if an R-peak existed at the center of the 500 ms reference signal section, the annotation was 1, otherwise 0.

The training data set was created by extracting an input/annotation pair from cECG, accelerometer and reference signals, and sliding the whole data by 1 ms (the sampling interval) to extract another pair, and so forth, and then the model training was performed using the data set.

4-4 Evaluation Method

The output from the trained 1D-CNN model, which takes on a fractional value, was thresholded to give either 0 (R-peak not found) or 1 (R-peak found) as the final output, and the performance was evaluated. For evaluation metrics, Precision, Recall, and *F*-score were used. Each metric was calculated as follows:

$$Precision = \frac{TP}{TP + FP} \times 100$$

$$Recall = \frac{TP}{TP + FN} \times 100$$

$$F\ score = \frac{2 \times Precision \times Recall}{Precision + Recall} \times 100$$

The definitions of TP, FP and FN are as follows:

TP : The number of R-peaks which was detected within the tolerance window of 20 ms of a true R-peak, and for which no other R-peaks are detected within the window. The true R-peaks are obtained from the reference signal.

FP : The number of R-peaks which was detected outside the tolerance window mentioned above, or for which other instances of such R-peaks are also detected within the same window (each detected instance will be counted).

FN : The number of true R-peaks which was not detected.

In addition, all R-peaks detected within the tolerance windows were evaluated in terms of the temporal root-mean-square error (RMSE) from the true R-peak locations.

Leave-one-subject-out cross validation was performed on the data and the average of the evaluation metrics over the 4 cross-validation evaluations was used as the R-peak detection performance of the model. In each of 4 cross-validation evaluations, data for two subjects, one subject, and one subject were used for the training data, the validation data, and the test data, respectively.

5. RESULTS AND DISCUSSION

5-1 Choice of the Accelerometer Channels to Use

The choice of the input channels of the accelerometer to be used can have an impact on the performance of the model. Here such choice was studied. The input channel candidates were cECG and accelerometer signals in three directions (Acc_x, Acc_y, Acc_z) as shown in Figure 4 obtained from the accelerometer.

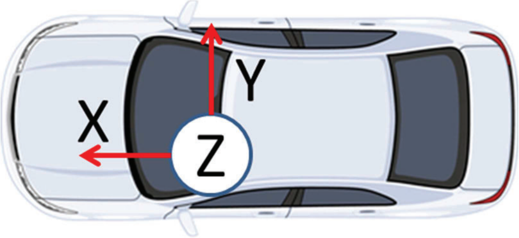


Figure 4. Three axis directions of the accelerometer

Table 2. Experimental results. Performance of R-peak detection on cECG single depending on the input signals to the 1D-CNN.

Input	Recall	Precision	*F*-score	RMSE
cECG	70.5	69.0	69.7	2.01
cECG, Acc_x	74.8	69.6	72.1	2.16
cECG, Acc_y	77.4	67.5	72.1	2.01
cECG, Acc_z	72.8	70.2	71.4	2.19
cECG, Acc_x, Acc_y, Acc_z	70.1	76.2	73.0	2.14

The results are shown in Table 2.

When all of Acc_x, Acc_y and Acc_z are used as input, the *F*-score is larger by 3% or more compared to the case where only cECG is used, and the RMSE is also a sufficiently small value of about 2 ms. Therefore, it can be said that the use of the accelerometer signals is advantageous for detecting R-peaks.

Figure 5 shows the waveforms of cECG, reference ECG, the vertical component of the accelerometer signal (Acc_z) and the output of the model.

In this example, a correlation is seen between the cECG and the accelerometer signal: where the noise intensity is large in cECG, the amplitude of the accelerometer signal is also large.

It is considered that the influence of the noise superimposed on the ECG signal can be canceled by using the accelerometer signal as an input of the 1D-CNN model, and as a result, erroneous R-peak detections can be suppressed.

5-2 Choice of the Input Data Length

The length of the data section will affect the performance of the model, and here such effect was studied. 1D-CNN model training was performed by choosing a different value for the input data section length, from 100 ms to 1500 ms. Note that the model inputs were cECG, Acc_x, Acc_y and Acc_z. The model output was the existence probability of the R-peak at the center of the input data section (for example, the output is the existence probability of the R-peak at 750 ms from the beginning of the data section if the input data length is 1500 ms), which was then thresholded to give either 0 or 1 as mentioned above. The results are shown in Table 3.

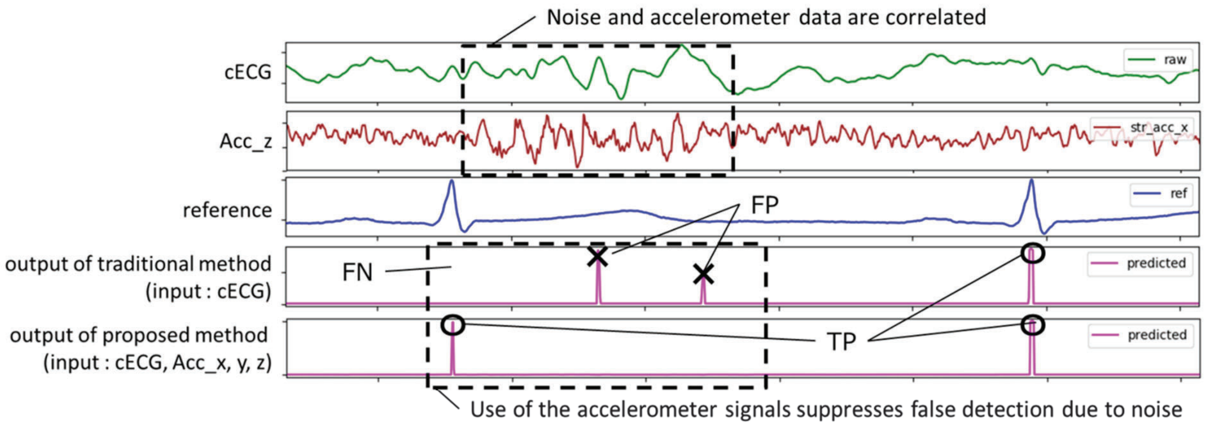
Generally, as the input data section length increases, the detection performance improves. The maximum *F*-score is achieved when the input data section length is 1100 ms, and thereafter, it starts to decrease. This may be due to the following reasons:

The RRIs of a resting healthy human are about 600 ms to 1000 ms, and the RRIs of our subjects are also roughly distributed in that range. Therefore, in the data used here, there are no other R-peaks in the range of around 1200 ms centered at one of the R-peaks. When the input data section length is 1100 ms (< 600 ms × 2), if an R-peak is located at the center of the data section (= output is 1), no other R-peaks exist in the interval. However, if the R-peak deviates from the center by several tens of ms or more, other R-peaks will enter the section. In other words, when the input data length is around 1100 ms, in addition to whether the R-peak is actually found at the center of the section (Criterion 1), whether no other R-peaks are found in the section (Criterion 2) can be used to determine whether we do have an R-peak at the center of the interval, so hence improving the accuracy.

If the input data section length is too short, the detection performance deteriorates due to insufficient information on the neighbor waves (P, Q, S, and T waves) in the input section, in which case the Criterion 1 becomes unreliable; and if the input data section length is too long (> 600 ms × 2), even if an R-peak is located at the center, other R-peaks can enter the input data section, and therefore the Criterion 2 becomes irrelevant (it won't be used). In either case, the detection performance is expected to deteriorate.

Table 3. Performance of R-peak detection depending on the input data window length (100 ~ 1500 ms).

Input data length	Recall	Precision	*F*-score	RMSE
100 ms	66.5	66.3	66.4	2.02
300 ms	71.6	71.8	71.7	2.26
500 ms	70.1	76.2	73.0	2.14
700 ms	72.7	73.4	73.0	1.98
900 ms	76.6	74.4	75.5	2.24
1100 ms	75.7	81.5	78.5	1.99
1300 ms	74.4	72.6	73.5	2.00
1500 ms	71.7	73.1	72.4	2.22



6. CONCLUSION

In this study, we proposed a method to accurately detect the driver's RRI in a running vehicle by the 1D-CNN, using the multi-channel inputs consisting of the accelerometer signals of the vehicle and the cECG of the driver, and discussed its results.

We confirmed that the detection performance was improved by more than 3% points in *F*-score by using the accelerometer signals as an input to the 1D-CNN together with the cECG data, acquired for a total of 20 minutes for 4 subjects while driving. In addition, the detection performance was improved as the input data section length was increased, and the maximum *F*-score of 78.5% is achieved when the input data section length is 1100 ms. Furthermore, under all conditions, a sufficiently small RMSE of about 2 ms was achieved, and the R-peaks were detected with sufficient accuracy to withstand HRV analysis.

Figure 5. Improvement of detection performance by adding accelerometer inputs

As a future work, we plan to add more training data to handle ECGs with diverse characteristics. We also plan to implement features to detect specific health problems, such as arrhythmias.

REFERENCES

[1] World Health Organization (2011). Global Atlas on cardiovascular disease prevention and control, viewed 3 Dec. 2019, < http://whqlibdoc.who.int/publications/2011/9789241564373_eng.pdf?ua=1>

[2] European Commission (2013). New Standards for Driving and Cardiovascular Diseases, viewed 3 Dec. 2019, <https://ec.europa.eu/transport/road_safety/sites/roadsafety/files/pdf/behavior/driving_and_cardiovascular_disease_final.pdf>

[3] U. Rajendra Acharya, K. Paul Joseph, N. Kannathal, C. Min Lim, Jasjit S. Suri (2006). Heart rate variability: a review, Med Bio Eng Comput, vol.44, pp.1031–1051.

[4] M. Kunugita, A. Harada, Y. Yamanaka, Y. Mizuno (2019). Study on noise generation mechanism from static electricity in capacitively coupled ECG sensor, 58th Annual Conference of Japanese Society for Medical and Biological Engineering, Okinawa, Japan.

[5] Lopez A, Richardson PC (1969). Capacitive Electrocardiographic and Bioelectric Electrodes, IEEE Transactions on Biomedical Engineering, BME-vol.16 (1), pp.99.

[6] Koivistoinen T, Junnila S, Va¨rri A, Ko¨o¨bi T (2004). A new method for measuring the ballistocardiogram using EMFi sensors in a normal chair. 26th IEEE EMBS conference, San Francisco, CA, USA.

[7] Guardo R, Charron G, Goussard Y, Savard P (1995). Contactless recordings of cardiac related thoracic conductivity changes. 17th IEEE EMBS conference, Montreal, Canada.

[8] Ichapurapu R, Jain S, John G, Monday T, Lie DYC, Banister R, Griswold J (2009). A 2.4 GHz non-contact biosensor system for continuous vital-signs monitoring.10th Annual IEEE wireless and microwave technology conference, Clearwater, FL, USA.

[9] M. Walter, B. Eilebrecht, T. Wartzek, S. Leonhardt (2011). The smart car seat: personalized monitoring of vital signs in automotive applications, Personal and Ubiquitous Computing archive, Vol.15 (7), pp.707-715.

[10] S. Leonhardt, A. Aleksandrowicz (2009). A Non-contact ECG monitoring for automotive application, 5th International workshop on wearable and implantable body sensor networks, Hong Kong, China.

[11] M. Kisohara, Y. Masuda, E. Yuda, J. Hayano (2019). Usefulness of Adaptive Correlation Filter for Detecting QRS Waves from Noisy Electrocardiograms. 2019 IEEE 1st Global Conference on Life Sciences and Technologies, Osaka, Japan.

[12] C. H. Antink, E. Breuer, D. U. Uguz, S. Leonhardt (2018). Signal-Level Fusion with Convolutional Neural Networks for Capacitively Coupled ECG in the Car, 2018 Computing in Cardiology Conference, Maastricht, Netherlands.

[13] T. Hirota, R. Fujita, M. Otake, Y. Nawa, K. Yamada, D. Kawamura (2019). R-wave detection from low S/N electrocardiogram using 1D convolutional neural network, 8th-aim-mtg of special interested group on artificial intelligence in medicine, Yokohama, Japan.

[14] Moody GB, Moody B, Silva I (2014). Robust Detection of Heart Beats in Multimodal Data: The PhysioNet/Computing in Cardiology Challenge 2014, Computing in Cardiology 2014, vol.41, pp.549–552.

VERIFICATION AND VALIDATION OF MACHINE LEARNING APPLICATIONS IN ADVANCED DRIVING ASSISTANCE SYSTEMS AND AUTOMATED DRIVING SYSTEMS

Chung-Jen Hsu
Bowhead Mission Solutions
USA

Paper Number 23-0048

ABSTRACT

The verification and validation processes of machine learning applications in advanced driving assistance systems or automatic driving systems are presented, and the processes are implemented by using the forward collision warning of pedestrian automatic emergency braking. Supervised learning is one of the machine learning branches using image datasets to train the deep neural network for detecting or identifying the target object or scenario in a vision-based application. The verification process consists of specifying the requirements of a safety functionality, identifying the target objects in the Operation Design Domain (ODD) and pre-crash scenarios, and evaluating the quality and quantity of images based on safety requirements, also the coverage of ODD and pre-crash scenarios. The validation process consists of designing test procedures based on the specified ODD and pre-crash scenarios, conducting a sufficient number of tests, recording the test results, and evaluating the test results based on specified metrics. Eight published pedestrian datasets from 2010 to 2020 are reviewed. Three datasets contain the raining condition, but no dataset had images collected during snowing days. Fog or smoke images are not available in all datasets, and the headlight condition is not addressed in all datasets. The 3 datasets containing pedestrians in the nighttime did not label the vehicle's headlight status as low or high beam. All reviewed datasets had no annotations of pre-crash scenarios that the subject vehicle is maneuvering or not. The validation of pedestrian detection uses the activation of forward collision warning as the evaluation metric. Eleven vehicles were tested in 4 pre-crash scenarios with different pedestrian orientations and speeds: the test pedestrian crossing from the nearside, crossing from the offside, stationary facing away, and walking away in front of the vehicle. The vehicle speed under test is 40 kph and the test pedestrian's speed is 5 or 8 kph. The light conditions are daytime, nighttime with low beam, and nighttime with high beam without streetlighting in a test track. The statistical test results show that some vehicles under test behave inconsistently when the test pedestrian is crossing or not crossing. Test results in the nighttime with high beam are similar to that of the daytime; however, the test results in the nighttime show significant variations compared with that of daytime. No trend or similarity can be found among all vehicles under test, the same vehicle may behave inconsistently under different light conditions and pedestrian orientations. Also, the pedestrian detection time is longer when the test pedestrian is not crossing for some vehicles. The vision-based machine learning application for the vehicle safety functionality reveals the underlying uncertainty of a deep neural network, and it results in the inconsistent performance in differentiated ODD conditions and pre-scenarios.

INTRODUCTION

Machine learning (ML) techniques have been widely used in the safety functions and vehicle control of Automated Driving Systems (ADS). ADS perform object and event detection and response consisting of monitoring the driving environment and execute appropriate responses to objects and events. The driving environment can also be referred to as Operational Design Domain (ODD) specifying the operating domains or conditions in which Advanced Driving Assistance Systems (ADAS) or ADS are designed to function safely. Object and event detections can be achieved by using cameras, radars or lidars to retrieve images for further processing. Supervised ML models can be used to identify vehicles, pedestrians, and other objects such as traffic signs, obstacles, and lane markings. Supervised learning is one of the ML paradigms being extensively applied for detecting objects through the training of a Deep Neural Network (DNN) with sufficient images of the target objects [1]. The detection accuracy depends on the quality and quantity of the training images and DNN modeling. Collecting and labeling images containing target objects are time consuming, and this effort is proportional to the numbers of object categories. Vehicles, pedestrians, cyclist, traffic signals, signs, lane markings, etc. are some objects to be identified in ADAS/ADS applications. If training images are not labeled correctly or they are unable to cover most of the target object categories, then the detection accuracy will not be sufficient for the safety requirements of ADAS/ADS even though the DNN modeling is impeccable. Scene semantic segmentation is to identify multiple objects and segment them as a relational group revealing a specific scenario in an image [2]. Pre-crash scenarios are crucial to safety applications to recognize the scene semantics in different ODD conditions. If an ADAS/ADS application can recognize driving scene semantics using a DNN, then the categories of scenes and related characteristics should also be examined for verifying the safety limitations of ADAS/ADS in the same manner as verifying object detections. The Verification and Validation (V&V) of ML applications in ADAS/ADS are not addressed comprehensively by using conventional engineering approaches as specified in automotive standards including ISO 26262 and 21448 [3]. One of the V&V challenges of ADAS/ADS safety applications is lack of transparency in ML development processes including the DNN modeling and training data. Due to the complex and proprietary characteristics of DNN modeling, it is difficult to verify its robustness by reviewing DNN's structures and algorithms; however, the validation can be achieved by measuring the level of detection accuracy. This study intends to tackle the V&V of ML applications in ADAS/ADS safety functionalities by verifying the training datasets and validating the performance from the safety perspective.

RELATED WORKS

Borg et al. [4] conducted a review of V&V for ML in the automotive industry. This study found a gap between current safety standards and contemporary ML-based safety-critical systems from the V&V perspective. Potential methodologies of V&V in ML applications can be categorized as: formal methods, control theory, probabilistic methods, process guidelines, and simulated test cases. The challenges are no clear certification processes of safety-critical systems with DNNs, a lack of transparency in ML processes, and concerns of the robustness and state-space explosion. The challenges essentially originate from the workflow of supervised ML that training data are fed into a DNN, and test data are used to validate whether the design requirements are fulfilled [5]. The insufficient robustness and out of scope state-spaces are caused by the lack of comprehensive coverage in training data or the design defect in a DNN. Depending on the design purposes of ML applications in ADAS/ADS, this drawback poses safety risks on scene identification, motion planning, decision making, vehicle control, or communication [6]. The first step of scene identification is to perceive the objects of interest that might result in a safety risk. The perception tasks of ADAS/ADS are implemented by using cameras, lidars, or other sensors. Sensors provide images of vehicles, pedestrians, cyclist, lane markings, etc. that had been collected and labeled as training datasets for developing ADAS/ADS applications. Yurtsever et al. [7] surveyed 18 driving datasets being used for ADS developments; however, only 6 of them covers various weather conditions in the daytime and nighttime. The insufficiency of ODD coverage for training datasets emerges as a safety risk. Burton [8] proposed to set criteria of selecting training data based on the semantic analysis of triggering conditions (pre-crash scenarios) or other causes of errors for safety assurance. Schwalbe and Schels [9] conducted a survey on methods for the safety assurance of ML-based systems and summarized that data representativity requirements including the scenario coverage, input space ontology, and experience collection can be used to validate ADAS/ADS functionalities. Willers et al. [10] proposed mitigation approaches to safety concerns including the data distribution's approximation of real world, data shifting over time, inadequate separation of tests and training data, and dependence on the labeling quality. Other safety concerns related to the DNN modeling are the brittleness of DNNs, unreliable output confidence information, unknown behavior in rare critical situations, and incomprehensible behavior. The mitigation approaches to addressing data concerns are the sensible data acquisition strategy, iterative analysis of test results, data labeling guidelines, continuous learning and updating, and data partitioning guidelines. Cheng et al. [11] measured the robustness, interpretability, completeness, and correctness of DNNs by metrics including the scenario coverage, neuron activations, neuron activation pattern, adversarial confidence loss, scenario-based performance degradation, interpretation precision, occlusion sensitivity covering, and weighted accuracy/confusion. Calculations of these metrics require analyses of DNNs attributes and corresponding images. Amershi et al. [12] also stated that ML components are more difficult to handle as distinct modules than traditional software components. ML models may be entangled with data in complex ways and experience non-monotonic erroneous behaviors. The validation of ADAS/ADS applications can be achieved by various testing methodologies to address issues of residual risks, including the pre-deployment road tests, closed course testing, full/simplified vehicle environment simulations, and subsystem simulations [13]. Residual risks are unexpected scenarios/environment, unexpected human driver behavior, degraded infrastructure, and road hazards. A direct measurement of the failure rate remains a viable approach to validate the ML applications in ADAS/ADS with the consideration of residual risks [8].

METHODS

Supervised learning is a paradigm of utilizing a large and representative set of labeled data to train a ML model. The training dataset is the crucial factor of determining the accuracy of object detection in ADAS/ADS applications. A vision-based system requires objects of interest in the training dataset. The rationale is without the objects of interest in the training dataset the probability of detecting the objects of interest can be close to zero, but not zero for false positives may exist in DNNs. The best practice of improving the detection accuracy is to provide high quality images in the training dataset and develop a decent DNN that can achieve a high detection rate. To ensure the detection accuracy in a vision-based ML application, the first step is to verify the training dataset that should have the required quality and quantity in the desired ODD conditions and pre-crash scenarios. The verification is the process of evaluating whether the training dataset meets the safety requirements.

Verification

The process of verifying a training dataset is shown in Figure 1. The first step is specifying the requirements of an ADAS/ADS safety functionality. This is comparable with the specification of software safety requirements in the design phase of software development as defined in ISO 26262 [14]. The objects in potential risks of collision need to be specified based on the safety requirements. Most common objects in the driving ODD are vehicles, pedestrians, motorcycle, cyclists, and other objects that may be struck by the subject vehicle. Table 1 lists the top-level categories of ODD classifications [15]. The images of a training dataset can be categorized according to physical infrastructure, operational constraints, objects, and environmental conditions. In addition, the semantics of images can be categorized based on the pre-crash scenario groups including control loss, road departure, animal, pedestrian, cyclist, lane change, opposite direction, rear-end, and crossing paths as listed in Table 2 [16]. Each image can be labeled according to the categories of ODD and pre-crash scenarios. Image labeling is a labor-intensive task that most existing ML datasets are labeled in the object level. A semantic level labeling task demands more human endeavors, so the automation is desirable for mitigating the cost and time of labor. Recent research of ADS started to work on the semantic scene identification [2, 17, 18, 19], these techniques can be applied to the labeling of ODD and pre-crash scenarios [20, 21]. After categorizing and calculating the quantity of images in categories of ODD and pre-crash scenarios, the distribution of images in each category can be reviewed and a reasonable inference can be made. For example, if no nighttime illumination of pedestrian images is available in the training dataset, then the detection rate of nighttime pedestrians will be low most likely. The number of images in a specified category can be an evaluation metric. Also, the target object's distribution in ODD and pre-crash categories indicates a sensible expectation of detection rates in those categories. High density categories may have a better detection rate; on the contrary, low density or no coverage categories may have a low or even zero detection rate. This information can also be the context of testing ODD conditions in the validation process. For example, the edge testing cases can be designed based on the rare conditions in the training dataset. The distribution of images in pre-crash scenarios is useful for providing the potential risk assessment across the coverage map. Different pre-crash scenarios represent differentiated viewing angles of the target object profiles. The vehicle profiles are different in the lane change, opposite direction, rear end, and crossing path scenarios. A pedestrian profile is different when they are crossing or not crossing (facing) a roadway. Lastly, the age of a training dataset may be a safety concern. When a dataset only contains outdated vehicles that may result in some new vehicles undetected.

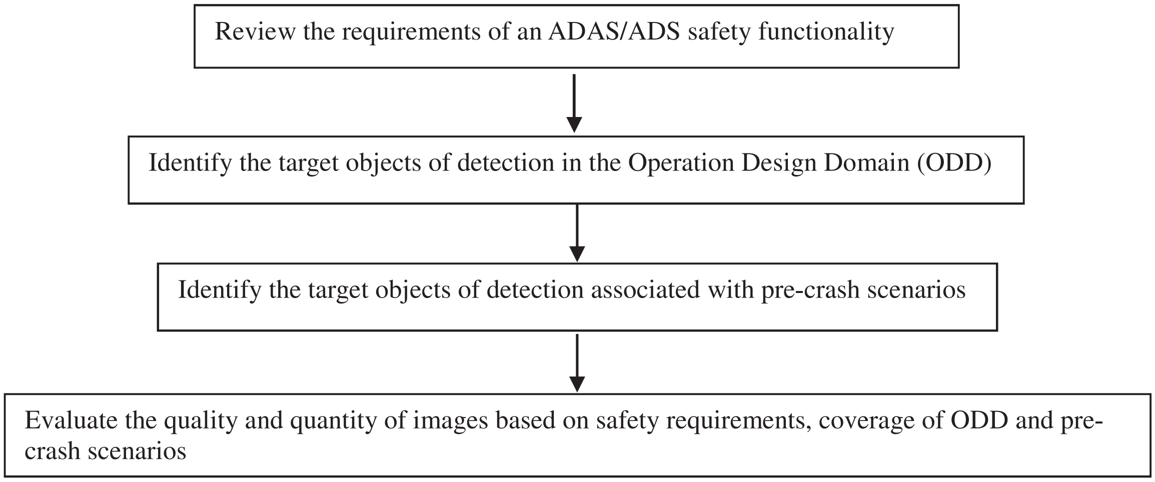

Review the requirements of an ADAS/ADS safety functionality
Identify the target objects of detection in the Operation Design Domain (ODD)
Identify the target objects of detection associated with pre-crash scenarios
Evaluate the quality and quantity of images based on safety requirements, coverage of ODD and pre-crash scenarios

Figure 1. The process flow of verifying a training dataset.

Table 1. ODD classification with top-level categories

ODD Element	**Conditions**	**Categories**
Physical Infrastructure	• Roadway Types • Roadway Surfaces • Roadway Edges • Roadway Geometry	• Divided/undivided highway, arterial, urban, rural, parking, bridges, multi-lane/single lane, managed lanes (HOV, reversible lanes), on-off ramps, one-way, private roads, intersections • Asphalt, concrete, unpaved • Lane markers, temporarily lane markers, shoulder, barriers, curb • Horizontal/vertical alignment (curves, hills), superelevation, lane width
Operational Constraints	• Speed Limit • Traffic Conditions	• High/low • Traffic density, others (emergency vehicles, construction, closed road, special event)
Objects	• Signage • Roadway Users • Obstacles/Objects	• Signs (stop, yield, pedestrian, railroad, school zone, etc.), traffic signals, crosswalks, railroad crossing, stopped buses, construction signage • Vehicle types (cars, light trucks, large trucks, buses, motorcycles,), stopped vehicles, pedestrians, cyclists • Animals, debris
Environmental Conditions	• Weather • Weather-induced Roadway Conditions • Particulate Matter • Illumination	• Precipitation, wind, snow, temperature • Standing water, flooded, icy, snow • Fog, smoke, smog, dust/dirt • Dark, streetlights, dawn/dusk, low sun angle, day light, headlights (low/high beam), oncoming vehicle lights

Table 2. Pre-crash scenarios and groups

Scenario Group	**Pre-Crash Scenarios**	**Subject Vehicle Maneuver**
Control Loss	• Control loss/maneuver • Control loss/no maneuver	• Maneuver: performing a maneuver (e.g., passing, turning, changing lanes) • No maneuver: driving straight or negotiating a curve
Road Departure	• Road edge departure/maneuver • Road edge departure/no maneuver	
Animal	• Animal/maneuver • Animal/no maneuver	
Pedestrian	• Pedestrian/maneuver • Pedestrian/no maneuver	
Cyclist	• Cyclist/maneuver • Cyclist/no maneuver	
Lane Change	• Turning/same direction • Parking/same direction • Changing lanes/same direction • Drifting/same direction	• Turn and cut across the path of another vehicle initially traveling in the same direction • Enter or leave a parked position and collide with another vehicle • Change and encroach into another lane other vehicle traveling in the same direction • Drift into an adjacent lane other vehicle traveling in the same direction
Opposite Direction	• Opposite direction/maneuver • Opposite direction/no maneuver	• Make a maneuver (e.g., passing) and encroach into another vehicle traveling in the opposite direction • Drift and encroach into another vehicle traveling in the opposite direction
Rear-End	• Rear-end/striking maneuver • Rear-end/Lead Vehicle Accelerating • Rear-end/Lead Vehicle Moving • Rear-end/Lead Vehicle Decelerating • Rear-end/Lead Vehicle Stopped	• Change lanes or pass another vehicle and closes in on a vehicle ahead in the same lane • Close in on an accelerating lead vehicle ahead in the same lane • Close in on a moving vehicle ahead in the same lane • Close in on a decelerating lead vehicle ahead in the same lane • Close in on a stopped lead vehicle ahead in the same lane
Crossing Paths	• Right turn into path • Right turn across path • Straight crossing paths • Left turn across path, lateral direction • Left turn into path • Left turn across path, opposite direction	• Turn right and into the same direction of another vehicle crossing from a lateral direction • Turn right and into the opposite direction of another vehicle crossing from a lateral direction • Go straight and collide with another straight crossing vehicle from a lateral direction • Turn left and cross the path of another vehicle traveling in the opposite direction from a lateral direction (left) • Turn left into the path of another vehicle traveling in the same direction from a lateral direction (right) • Turn left and cross the path of another vehicle traveling in the opposite direction

Validation

Validation is the process of evaluating the degree to which a ML model/application and its data can provide an accurate result of the intended uses. Essentially, the validation of an ADAS/ADS application can be implemented after the verification of its training dataset that reveals the coverage and distribution across the spectrum of ODD and pre-crash scenarios. Depending on the design specification of a safety function, the validation tests can be conducted focusing on the selected ODD and pre-crash scenarios. Also, validation test procedures can be designed based on the historical crash data. High crash frequency scenarios may be tested with a higher priority and number of test runs. Well-design test procedures should be able to address safety concerns including rare critical situations, unreliable confidence information of DNN output, and brittleness of DNNs [10]. An ADAS/ADS safety functionality consists of software and hardware working together to achieve the goal of crash avoidance. The DNN is a part of software processing images from sensors (cameras) and generates the detection results. The uncertainty of a DNN output and the risk of hardware glitches result in the safety performance of a vehicle under test. Broken sensors or alarming devices result in no alarm that can be easily distinguish from software malfunctions in a few test runs. A vehicle-level test is feasible to validate the DNN of a safety application excluding the hardware failure. Safety metrics for evaluating the DNN performance also need to include the response time in addition to the pass/fail metric. Sufficient test runs are needed to collect data for calculating the reaction time and figuring out the boundary or capability of the DNN under test. The flowchart of validating a ML safety application is shown in Figure 2.

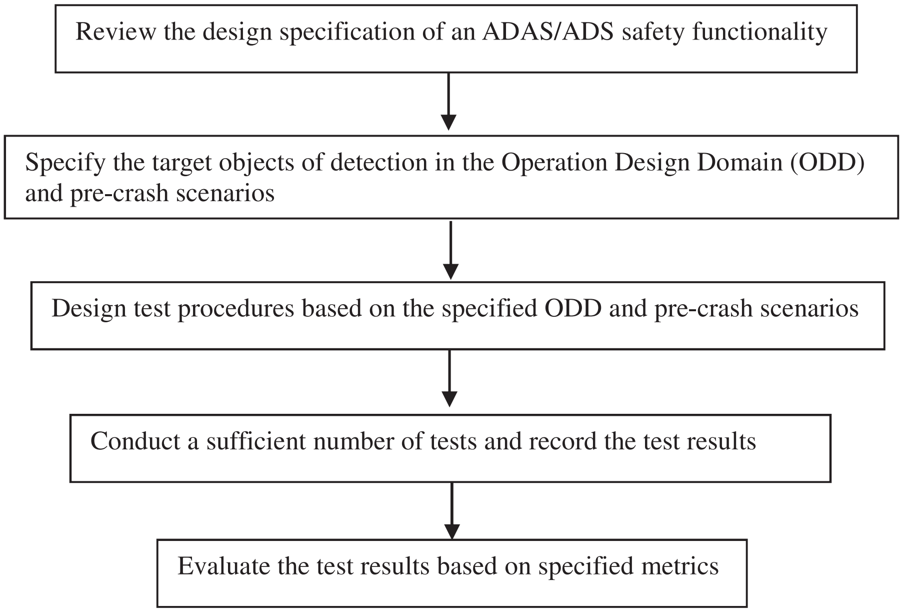

Review the design specification of an ADAS/ADS safety functionality
Specify the target objects of detection in the Operation Design Domain (ODD) and pre-crash scenarios
Design test procedures based on the specified ODD and pre-crash scenarios
Conduct a sufficient number of tests and record the test results
Evaluate the test results based on specified metrics

Figure 2. The process flow of validating an ADAS/ADS functionality with ML applications.

VERIFICATION OF TRAINING DATA

In consideration of selecting an ADAS/ADS safety application for the proposed methodology, the pedestrian crash avoidance is appropriate for a sensible reason. Pedestrian detection approaches are feature-based or ML-based or hybrid using different image processing algorithms [22, 23, 24]. Vehicle detections may use sensors such as lidar and radar along with cameras to provide inputs to the detection software, but the pedestrian detection mostly relies on cameras. The rationale is when the training dataset containing pedestrians in the specified ODD conditions and scenarios, the DNN may be able to detect the pedestrian at risk successfully. By inspecting the training dataset in line with the proposed verification process, the boundaries or limitations of pedestrian detection can be identified.

Safety Functionality Requirements

Pedestrian Automatic Emergency Braking (PAEB) is a safety function of ADAS. A PAEB system consists of cameras, Forward Collision Warning (FCW), and last moment automatic braking to prevent a collision with pedestrians. FCW can be used as an indicator of whether a pedestrian at risk is detected or not. FCW is designed to warn the driver to maneuver or brake as early as possible.

Detection Object in Operation Design Domain

The pedestrian detection is to identify humans in the environment where the subject vehicle is traveling with potential collision risks. After reviewing the ODD conditions as listed in Table 1, roadway types, weather, particulate matter, and illumination are directly related to the performance of pedestrian detections. Most pedestrians appear at intersections, arterials, and parking lots in urban areas, less may be seen on all types of roadways in rural areas. Verifying the roadway distribution of pedestrian images can be beneficial to understating the background of the training dataset. Although the pedestrian detection capability and performance may not be correlated to the background of road types, such information can be used for the design of edge test scenarios. For example, a pedestrian is walking on the freeway shoulder.

Detection Objects in Pre-crash Scenarios

Two pre-crash scenarios of pedestrians are considered in Table 2 that the subject vehicle is maneuvering or not. When a vehicle is making a turn at an intersection where pedestrians are crossing, the viewing angle from the subject vehicle to a crossing pedestrian is changing in the process of turning. Ideally, the training dataset is expected to contain a variety of pedestrian profiles from various viewing angles.

Verification of Training Datasets

Eight pedestrian datasets are reviewed as listed in Table 3. The publication year, number of images, number of pedestrians, image resolution, pedestrian annotation (labeling), camera setup, and data collection areas reveal the background information of each dataset. The recent published datasets are reviewed since 2010 for aged pedestrian datasets may not have the sufficient quantity, resolution, and annotation to be used for training recent pedestrian detectors. One dataset labeled pedestrian images as the full, part, or just head of a pedestrian depending on the level of occlusion. Most datasets had a camera installed on the vehicle recording videos of pedestrians, two datasets collected pedestrians or human images from internet sources. The data collection area provides the information of where the pedestrian images were obtained.

Table 3. Dataset characteristics

Dataset	Year	# Image	#Pedestrian	Resolution	Annotation	Setup	Area
Caltech [25]	2010	250k	289k	640*480	Full, body	Vehicle	LA metropolitan
KITTI [26]	2012	15k	9k	1240*376	Full	Vehicle	Mid-size city, rural areas
CityPersons [27]	2017	5k	35k	2048*1024	Full, body	Vehicle	27 cities, Germany
CrowdHuman [28]	2018	24k	552k	-	Full, body, head	Internet images	40 cities worldwide
NightOwls [29]	2018	281k	56k	1024*640	Full	Vehicle	7 cities, 3 countries in Europe
EuroCity [30]	2019	47k	219k	1920*1024	Full	Vehicle	31 cities, 12 countries in Europe
TJU-DHD [31]	2020	75k	373k	1624*1200 2560*1440	Full, body	Vehicle, phone	Road, off road (campus)
WiderPerson [32]	2020	13k	39k	1400*800	Full	Internet images	

Table 4 lists the verification result of pedestrian datasets based on the ODD conditions and pre-crash scenarios. Ideally, the dataset verification process should have a labeling tool that is able to identify the ODD's condition/category and pre-crash scenario. However, such a tool is not available currently and the manual labeling of thousands of images is a huge task. The developers of ML pedestrian detection algorithms are expected to verify their training datasets with the consideration of ODD conditions and pre-crash scenarios. The review of datasets in this paper is based on the revealed information from the dataset publications. Three datasets contain raining weather conditions, but no dataset mentioned images collected during snowing days. Snow, fog, or smoke images are not available in all datasets, for such conditions may be rare during the data collection period and were not mentioned in the publications. The headlight condition is not addressed in all datasets, the 3 datasets containing pedestrians at night did not annotate the vehicle's headlights as low or high beam. When the data collection vehicle was traveling at night, it would be a reasonable inference that both low and high beams had been used. As for the pre-crash scenarios as the subject vehicle is maneuvering or not, it is most likely the vehicle had been making turns, changing lanes, and keeping straight with one or more pedestrians ahead in the period of data collection.

Table 4. Review of datasets based on ODD and pre-crash scenarios

Condition	**Category**	**Dataset**							
		Caltech	KITTI	CityPersons	Crowd Human	Night Owls	Euro City	TJU-DHD	Wider Person
Roadway	Intersection	✓	✓	✓	✓	✓	✓	✓	✓
	Arterial	✓	✓	✓	✓	✓	✓	✓	✓
Weather	Sunny/cloudy	✓	✓	✓	✓	✓	✓	✓	✓
	Rain					✓	✓	✓	
	Snow								
Particulate Matter	Fog/Smoke/Dust								
Illumination	Day	✓	✓	✓	✓		✓	✓	✓
	Night streetlight					✓	✓	✓	
	Headlight low beam					+	+	+	
	Headlight high beam					+	+	+	
Maneuver	Turning/changing lane	✪	✪	✪		✪	✪	✪	
Not Maneuver	Straight	✪	✪	✪		✪	✪	✪	

+: The dataset may contain low or high beam headlight images, but they are not identified specifically.

✪: The data collection vehicle should have both pre-crash scenarios in the data collection process.

VALIDATION OF SAFETY APPLICATIONS

The validation process as elucidated in Figure 2 is implemented in the following. The ML safety application is PAEB, and FCW uses the pedestrian detection to trigger an alarm. Vehicle level tests of PAEB were conducted in 2020 under the supervision of National Highway Traffic Safety Administration (NHTSA). Eleven vehicles equipped with FCW in PAEB of model year 2020 from 10 manufacturers including 4 sedans, 5 SUVs, 1 minivan, and 1 pickup truck were tested. The validation of pedestrian detection is to test whether FCW is activated in the selected ODD conditions and pre-crash scenarios.

Design Specifications of FCW in PAEB

FCW's operational speed ranges and limitations of 11 tested vehicles are collected from owner manuals and listed in Table 5. Eight vehicles' operational speeds are not higher than 50 mph, 3 of them are higher than 50 mph. The lower end of operational speeds is from 3 to 7 mph. The negative factors of vision-based pedestrian detection algorithms are summarized from the operational limitations listed in the owner manuals. Just like the limitations of human eyes, extreme light, air, and inclement weather conditions hinder the capability of pedestrian detections. In addition, the shape, movement speed, color, posture, and clothing of pedestrians are sensitive to detection results. Occlusions or carrying objects are negative factors to the detection accuracy, and the detectable pedestrian height is from 1 to 2 meter for some vehicles. Although it is not mentioned in all owner manuals, the ideal detection can only be achieved in the straight and flat road alignments.

Table 5. FCW operation speed range and limitations of tested vehicles

Vehicle	Operation Speed (mph) Low	High	Limitations
1	6	50	Curves, heavy fog rain, snow, dark, occlusion, glare, light variation/reflections
2	6	50	Curves, heavy fog rain, snow, dark, occlusion, glare, light variation/reflections
3	3	75	Not available
4	3	62	Height:1-2 m, groups, occlusion, unusual shape, movement (running)
5	5	45	Not walking upright, sudden appearance, small, clothes blend into background, too bright or dark, inclement weather
6	3	37	Less than 1m, carrying luggage, severe weather
7	4	43	Up to 50 mph for moving pedestrians, 43 mph for stationary pedestrians, snow, rain, fog, glare, sudden appearance, occlusion, blend into background, special clothing or object, tight curve
8	6	37	Small children, pedestrians on wheelchair/skateboard, not upright, darkness, strong light caused pedestrian in shadow, sudden change in brightness, occlusion, carrying luggage
9	7	100	1-2 m, in a group, next to obstacle, using umbrella, similar clothing color to background, carrying luggage, not upright, dark, sudden appearance, inclement weather, strong light from the front, dust, smoke, steam, steep up/down hill, darkness
10	7	50	1-2 m, abrupt appearance, not directly in front, near obstacle, occlusion, strong light, same color in the surrounding, oversize clothing, moving fast, not upright, pushing an object, inclement weather, steam, smoke, darkness, abrupt changing brightness, curve
11	3	50	Shorter than 0.8m, clothing covering body contour, poor background contrast, carrying a large object

ODD and Pre-crash Scenarios of Pedestrian Detection Tests

The test ODD conditions and pre-crash scenarios can be specified after reviewing the limitations of vehicles under test. A test process may start from easy or most common ODD conditions and pre-crash scenarios, then increase the difficulty level gradually depending on the testing requirements. Light conditions and pedestrian movement orientations are two major test variables including day and night under low and high beams, crossing from the nearside or offside, or walking toward/backward in front of the vehicle. The selected NHTSA test scenarios [33] are:

- S1b: the vehicle encounters a crossing adult from the nearside (closest to the curb)
- S1e: the vehicle encounters a crossing adult running from the offside (closest to the center of the road)
- S4a: the vehicle encounters a stationary adult on the nearside of the road facing away
- S4c: the vehicle encounters an adult on the nearside of the road walking in the same direction

These 4 test scenarios have both day and night test results for the validation.

Test Procedures

A test pedestrian mannequin with swinging arms is used to simulate an adult pedestrian whose speed and direction can be controlled. The vehicle under test is driven at the specified speed approaching the test mannequin moving in the orientation as defined in the test scenarios. The test site has no overhead signs or other significant structures to cause occlusions. Each trial was conducted without other vehicles, obstructions, or stationary objects within one lane width on either side of the driving lane. All tests are conducted without inclement weather conditions such as fog, smoke, or ash. Also, the daytime tests were conducted with good visibility without direct sunlight or glare. The nighttime tests were conducted without streetlighting. The speeds and orientations of the test vehicle and pedestrian are listed in Table 6.

Table 6. Vehicle and pedestrian speeds and orientations under tests

Test Scenario	Vehicle Speed (kph)	Pedestrian Speed (kph)	Pedestrian Orientation
S1b	40	5	Crossing nearside
S1e	40	8	Crossing offside
S4a	40	0	Stationary facing away
S4c	40	5	Walking away

Evaluation of Test Results

The accuracy of pedestrian detections depends on various limitation factors as elucidated in Table 5. The test scenarios are designed to exclude most unfavorable factors and focus on the most common ODD conditions and pre-crash scenarios. The pedestrian profile or shape can be different from the camera's viewing angle in the daytime or nighttime. The low or high beam light is the major source of light on the test pedestrian. The performance of pedestrian detections can be validated by comparing the detection results. The statistical test — Fisher's exact test [34] is used to evaluate whether the pedestrian detection algorithm is independent of a pedestrian's crossing or not under 3 light conditions. In other words, the test is to find out whether the detection algorithm's performance is the same under various profile conditions. The null hypothesis is that the pedestrian detection algorithm is independent of the test pedestrian's profile. A two-tale P value is used to determine whether the null hypothesis can be rejected or not. When the P value is greater than the significance level (α) 0.05, there is no evidence to reject the null hypothesis. The independence means the pedestrian detection algorithm behaves similarly when the pedestrian's profile varies. Alternatively, the detection algorithm behaves differently when the pedestrian's profile varies. Lastly, the detection time in terms of time to collision is summarized for comparing the performance of all vehicles under test.

Crossing versus Not Crossing Test scenarios S1b and S1e are crossing pedestrians from the nearside or offside, and test scenarios S4a and S4c are not crossing pedestrians but walking or standing in front of the vehicle under test. The test results of FCW are either a warning activated or not. The totals of warnings and no warnings for crossing and not crossing scenarios in the daytime are listed for each vehicle under test in Table 7. Fisher's exact tests are also conducted to evaluate the pedestrian detection of each vehicle. Most vehicles are able to detect the test pedestrian consistently whether it is crossing or not. Vehicles 4, 5, and 7 under test behaved inconsistently when they are detecting the test pedestrian in different orientations.

Table 7. Validation of pedestrian detection for crossing and not crossing scenarios in daytime

Vehicle	Crossing		Not Crossing		Fisher's Test	
	Warning	No Warning	Warning	No Warning	P Value	Null Hypothesis
V1	11	0	10	0	1	Not reject
V2	7	1	8	0	1	Not reject
V3	8	0	5	0	1	Not reject
V4	0	6	6	0	0.002	Reject
V5	10	0	4	4	0.023	Reject
V6	8	1	10	0	0.474	Not reject
V7	11	0	1	4	0.003	Reject
V8	10	0	8	0	1	Not reject
V9	11	0	10	0	1	Not reject
V10	10	0	10	0	1	Not reject
V11	10	0	6	0	1	Not reject

The totals of warnings and no warnings for crossing and not crossing scenarios in the nighttime (low beam) are listed for each vehicle under test in Table 8. Fisher's exact test result shows only vehicle 5 did not behave consistently when detecting the test pedestrian crossing or not. However, the variations of warnings and no warnings increase as compared to the daytime test results.

Table 8. Validation of pedestrian detection for crossing and not crossing scenarios in nighttime (low beam)

Vehicle	Crossing		Not Crossing		Fisher's Test	
	Warning	No Warning	Warning	No Warning	P Value	Null Hypothesis
V1	8	0	10	0	1	Not reject
V2	3	3	5	1	0.546	Not reject
V3	3	3	6	0	0.182	Not reject
V4	0	7	4	3	0.07	Not reject
V5	8	0	0	4	0.002	Reject
V6	0	6	0	5	1	Not reject
V7	11	0	3	1	0.267	Not reject
V8	0	5	0	6	1	Not reject
V9	8	0	6	0	1	Not reject
V10	10	0	10	0	1	Not reject
V11	3	3	4	3	1	Not reject

The totals of warnings and no warnings for crossing and not crossing scenarios in the nighttime (high beam) are listed for each vehicle under test in Table 9. Fisher's exact test result shows 3 vehicles (4, 5, and 8) did not behave consistently when detecting the test pedestrian crossing or not. The test results are similar to the daytime test results.

Table 9. Validation of pedestrian detection for crossing and not crossing scenarios in nighttime (high beam)

Vehicle	Crossing		Not Crossing		Fisher's Test	
	Warning	No Warning	Warning	No Warning	P Value	Null Hypothesis
V1	7	0	8	0	1	Not reject
V2	9	0	6	0	1	Not reject
V3	11	0	10	0	1	Not reject
V4	0	6	5	1	0.015	Reject
V5	10	0	3	4	0.015	Reject
V6	0	5	5	3	0.075	Not reject
V7	10	0	11	0	1	Not reject
V8	5	5	0	7	0.044	Reject
V9	10	0	10	0	1	Not reject
V10	9	1	10	0	1	Not reject
V11	8	0	12	0	1	Not reject

Only vehicle 5 is not able to detect the test pedestrian consistently under crossing or not crossing conditions in daytime, nighttime with low beam, and nighttime with high beam.

Detection Time Table 10 shows the detection times of all vehicles under test in 4 test scenarios and 3 light conditions. In the daytime, the average detection times of vehicles 1, 2, 6, 8, 9, 10, and 11 are longer when the test pedestrian is not crossing. In the nighttime (low beam), the average detection times of vehicles 2 and 10 are longer when the test pedestrian is not crossing. In the nighttime (high beam), the detection times of vehicles 3, 7, 9, 10 and 11 are longer when the test pedestrian is not crossing. On average, the vehicles under test take a longer time to detect the test pedestrian when it is not crossing.

Table 10. Averages of FCW detection times of vehicles under test

Vehicle	Day				Night (low beam)				Night (high beam)			
	S1b	**S1e**	**S4a**	**S4c**	**S1b**	**S1e**	**S4a**	**S4c**	**S1b**	**S1e**	**S4a**	**S4c**
V1	1.16	0.66	1.27	1.66	1.07	0.63	1.07	1.37	1.22	0.56	1.07	1.64
V2	0.40	1.14	2.04	2.11	0.61	0.30	0.78	0.73	1.58	1.09	0.79	0.33
V3	1.53	1.07	0.66	1.64	*	1.06	0.66	1.21	1.42	1.26	1.49	1.47
V4	*	*	1.95	2.26	*	*	1.95	0.63	*	*	1.54	2.48
V5	1.10	0.92	0.91	0.35	0.97	0.77	*	*	1.02	0.76	*	1.57
V6	1.09	1.15	1.42	1.68	*	*	*	*	*	*	*	1.70
V7	0.76	0.9	1.12	*	0.84	0.93	*	0.34	0.78	0.89	2.08	1.64
V8	1.53	1.43	1.07	1.75	*	*	*	*	1.64	*	*	*
V9	1.48	1.47	2.66	2.57	0.97	0.75	0.85	1.19	1.30	0.69	1.63	2.10
V10	1.88	1.10	2.23	2.27	1.41	1.08	2.23	2.28	1.71	0.93	2.25	2.28
V11	1.65	1.29	2.04	1.84	0.18	0.06	*	0.38	1.54	1.25	2.01	2.09
Average	1.26	1.11	1.64	1.81	0.86	0.70	1.27	1.02	1.36	0.93	1.61	1.73

*: no data. Unit: second.

CONCLUSIONS

The verification and validation methodology of ML applications for safety functionalities in ADAS/ADS is presented, and an example of FCW in PAEB is demonstrated. The verification of training data provides insights and potential weaknesses of a ML application from the safety perspective in terms of ODD conditions and pre-crash scenarios. Most current pedestrian datasets are lack of inclement weather, weak illumination, air particulate matter, and vehicle/pedestrian maneuvering annotations. The validation process follows the lead of the verification result that the vehicle's headlight and background illumination conditions are needed to be tested under different pedestrian pre-crash scenarios. The test results show that some vehicles under test behave inconsistently when the test pedestrian is crossing as compared to not crossing. Test results in the nighttime with high beam headlight is similar to that of the daytime; however, the test results in the nighttime show significant variations compared with that of daytime. No trend or similarity can be found among all vehicles under test, the same vehicle may behave inconsistently under different light conditions and pedestrian orientations. Also, the pedestrian detection time is longer when the test pedestrian is not crossing on average. The vision-based ML application for the vehicle safety functionality reveals the uncertainty of a DNN, and it results in the inconsistent performance under differentiated ODD conditions and pre-scenarios.

REFERENCES

[1] Kuutti S., Bowden R., Jin Y., Barber F., Fallah S. "A Survey of Deep Learning Applications to Autonomous Vehicle Control." arXiv:1912.10773v1; 2019.

[2] Feng D., Haase-Schutz C., Rosenbaum L. et al. "Deep Multi-modal Object Detection and Semantic Segmentation for Autonomous Driving: Datasets, Methods, and Challenges." arXiv:1902.07830v4; 2020.

[3] Mohseni S., Pitale M., Singh V., Wang Z. "Practical Solutions for Machine Learning Safety in Autonomous Vehicles." arXiv:1912.09630; 2019.

[4] Borg M., Englund C., Wnuk K., et al. 'Safely Entering the Deep: A Review of Verification and Validation for Machine Learning and a Challenge Elicitation in the Automotive Industry." Journal of Automotive Software Engineering. 2019; 1(1), 1–19.

[5] Pullum L. "Verification and Validation of Systems in Which AI is a Key Element". SEBoK Editorial Board. 2021. Accessed 5/11/2022. www.sebokwiki.org.

[6] Elallid B. B., Benamar N., Hafid A. S., Rachidi T., Mrani N. "A Comprehensive Survey on the Application of Deep and Reinforcement Learning Approaches in Autonomous Driving." Journal of King Saud University – Computer and Information Sciences; 2022. https://doi.org/10.1016/j.jksuci.2022.03.013.

[7] Yurtsever E., Lambert J., Carballo A., Takeda K. "A Survey of Autonomous Driving: Common Practices and Emerging Technologies." IEEE Access, 2020; 8: 58443–58469.

[8] Burton S. "A causal model of safety assurance for machine learning." 2022; arXiv:2201.05451v1.

[9] Schwalbe G., Schels M. "A Survey on Methods for the Safety Assurance of Machine Learning Based Systems."

10th European Congress on Embedded Real Time Software and Systems; 2020, Toulouse, France.
[10] Willers O., Sudholt S., Raafatnia S., Abrecht S. “Safety Concerns and Mitigation Approaches Regarding the Use of Deep Learning in Safety-Critical Perception Tasks.” arXiv:2001.08001v1; 2020.
[11] Cheng C-H, et al. “Towards Dependability Metrics for Neural Networks.” arXiv: 1806.02338v2; 2018.
[12] Amershi S. et al. “Software Engineering for Machine Learning: A Case Study.” IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Practice; 2019, pp. 291-300, doi: 10.1109/ICSE-SEIP.2019.00042.
[13] Koopman P., Wagner M. “Toward a Framework for Highly Automated Vehicle Safety Validation.” 2018 SAE World Congress, SAE 2018-01-1071.
[14] Salay R., Queiroz R. and Czarnecki K. “An Analysis of ISO 26262: Using Machine Learning Safely in Automotive Software.” arXiv :1709.02435v1; 2017.
[15] Staplin, L., Mastromatto, T., Lococo, K. H., Kenneth W. Gish, K. W., & Brooks, J. O. “The effects of medical conditions on driving performance (Report No. DOT HS 812 623).” Washington, DC: National Highway Traffic Safety Administration, 2018.
[16] Swanson, E., Foderaro, F., Yanagisawa, M., Najm, W. G., & Azeredo, P. “Statistics of light-vehicle pre-crash scenarios based on 2011-2015 national crash data (Report No. DOT HS 812 745).” Washington, DC: National Highway Traffic Safety Administration, 2019.
[17] Dvornik N., Mairal J., Schmid C. “On the Importance of Visual Context for Data Augmentation in Scene Understanding.” arXiv:1809.02492v3; 2019.
[18] Rahman Q. M., Sunderhauf N., Corke P., Dayoub F. “FSNet: A Failure Detection Framework for Semantic Segmentation.” arXiv:2108.08748v2; 2021.
[19] Cheng C.-H., Knoll A., Liao H.-C. “Safety Metrics for Semantic Segmentation in Autonomous Driving.” arXiv:2105.10142v2; 2021.
[20] Gyllenhammar M. et al. “Towards an Operational Design Domain That Supports the Safety Argumentation of an Automated Driving System.” Proceeding of the 10th European Congress on Embedded Real Time Software and Systems, Toulouse, 2020.
[21] Riedmaier S., Ponn T., Ludwig D., Schick B., Diermeyer F. “Survey on Scenario-Based Safety Assessment of Automated Vehicles.” IEEE Access 2020; 8: 84756–84777.
[22] Ragesh N. K., Rajesh R. “Pedestrian Detection in Automotive Safety: Understanding State-of-the-Art.” IEEE Access. 2019; 7: 47864–47890.
[23] Cao J. et al. “From Handcrafted to Deep Features for Pedestrian Detection: A Survey.” IEEE Transactions on Pattern Analysis and Machine Intelligence. 44(9):4913-4934; 2022.
[24] Hasan I. et al. “Pedestrian Detection: Domain Generalization, CNNs, Transformers and Beyond.” arXiv:2201.03176v2, 2022.
[25] Dollar P., Wojek C., Schiele B., and P. Perona. “Pedestrian detection: An evaluation of the state of the art.” IEEE Transactions on Pattern Analysis and Machine Intelligence, 34(4):743–761, 2010.
[26] Geiger A., Lenz P., and R. Urtasun. “Are we ready for autonomous driving? the kitti vision benchmark suite.” Proc. IEEE Conference on Computer Vision and Pattern Recognition, 2012.
[27] Zhang S., Benenson R., and B. Schiele. “Citypersons: A diverse dataset for pedestrian detection.” Proc. IEEE Conference on Computer Vision and Pattern Recognition, 2017.
[28] Shao S. et al. “Crowdhuman: A benchmark for detecting human in a crowd.” arXiv:1805.00123, 2018.
[29] Neumann L. et al. “Nightowls: A pedestrians at night dataset.” Proc. Asian Conference on Computer Vision, 2018.
[30] Braun M., Krebs S., Flohr F., and D. M. Gavrila. “Eurocity persons: A novel benchmark for person detection in traffic scenes.” IEEE Transactions on Pattern Analysis and Machine Intelligence, 41(8):1844–1861, 2019.
[31] Pang Y., et al. “Tju-dhd: A diverse high-resolution dataset for object detection.” IEEE Transactions on Image Processing, 2020.
[32] Zhang S. et al. “Widerperson: A diverse dataset for dense pedestrian detection in the wild.” IEEE Transactions on Multimedia, 22(2):380–393, 2020.
[33] Pedestrian automatic emergency brake system confirmation test (working draft). Washington, DC: National Highway Traffic Safety Administration, 2019.
[34] Dowdy S., Wearden S., and D. Chilko. “Statistics for Research” 3rd Edition, 2004.

EVALUATION APPROACH FOR MACHINE LEARNING CONCEPTS IN OCCUPANT PROTECTION BASED ON MULTI-ATTRIBUTE DECISION MAKING

Franz Plaschkies
Technische Hochschule Ingolstadt
Germany
Ketlen Possoli
Federal University of Santa Catarina; Technische Hochschule Ingolstadt
Brazil; Germany
Ondrej Vaculin
Technische Hochschule Ingolstadt
Germany
Axel Schumacher
University of Wuppertal
Germany
Pedro de Andrade Junior
Federal University of Santa Catarina
Brazil

Paper Number 23-0055

ABSTRACT

The systems for occupant protection in passive vehicle safety are primarily developed with single statistical representations of humans, so-called Anthropomorphic Test Devices (ATDs). Unfortunately, those ATDs cover additional features like age and body shape insufficiently during development. Augmenting finite element simulations with a metamodel trained by machine learning is promising to overcome this barrier. However, the database design, the machine learning architecture, and the requirements for quality and robustness influence each other. Therefore, objective criteria must be defined to compare the alternatives taking cost and benefit aspects under changing preferences into account. Having complex criteria can be framed as a multi-attribute decision-making problem. This paper's objective is the development of a transparent assessment scheme for virtual statistical simulation for rapid vehicle occupant safety assessment using supervised learning.

PROMETHEE is selected as an appropriate decision-making approach. A process, consisting of a sequential definition of the criteria leading to the final assessment, is proposed to adapt the method in this paper's domain. The methodology is tested on sample alternatives, generated using a calibration-type machine learning architecture and data from finite element simulations. The original PROMETHEE algorithm cannot handle a vast number of alternatives. Since, typically, numerous alternatives occur during the development of a machine learning application, a sorting-based modification is implemented.

Finally, the findings are discussed, and recommendations for related use cases are given. The proposed method seems applicable to the described domain and near-related ones. Moreover, multiple tendencies between an alternative's parameters and rank can be identified in the test samples.

INTRODUCTION

In the recent years, passive vehicle safety has been dominated by the increasing virtualisation of assessment methods. Historically, crash tests are performed with real prototypes. A single virtual, physical simulation utilising, e. g. Finite Element Analysis (FEA), comes with significantly lower cost, higher flexibility, and an unmatched insight into the physical processes. However, the virtual simulations must fit the reality sufficiently, which makes extensive validation necessary. The degree of model detail and computational effort has been increased to fulfil the demand for trustworthy models. Nowadays, an industrial simulation on state-of-the-art hardware takes hours to days. Multiple developments led to the need for further acceleration of virtual methods: (i) shorter product cycles require rapid assessment; (ii) increased parameter spaces make more efficient methods for a sufficient assessment necessary; (iii) Euro NCAP recently proposed in [1] scenario-based virtual testing; (iv) the development in autonomous driving will introduce a broad range of allowed sitting positions and activities during driving, as stated by Östling et al. in [2]; (v) the population of vehicle occupants is significantly more diverse than it was when the anthropometrics for the state-of-the-art crash test dummies were developed, as concluded by Reed et al. in [3] and Wang et al. in [4]. Those dummies, so-called anthropomorphic test devices (ATDs), are technical measuring devices. They are the 5th, 50th, and 95th percentile representations of the North American population in the 1970s [5].

This paper proposes a method to develop a transparent assessment scheme for virtual statistical simulation for rapid vehicle occupant safety assessment using supervised learning. The methodology was tested on vehicle occupant safety assessment, specifically on the front crash case for passengers. The data originated from a simplified 2D FEA-model. The machine learning architecture contained a calibration approach introduced by Plaschkies et al. in [6] with supervised learning techniques.

Next to other influences, the above-described development sparked various publications of machine learning applications in the passive safety assessment, as summarised by Plaschkies et al. in [5]. The identified studies focused on prediction quality metrics like accuracy to assess and compare their investigated approaches. However, machine learning notably depends on the amount and quality of data leading to complex metrics with interacting parameters. Therefore, the trade-off between data generation costs and their value for the method must be represented. Approaches from Multi-Criteria Decision Making (MCDM) can provide a transparent way to compare different alternatives regarding multiple criteria to solve a particular problem.

STATE-OF-THE-ART BASED SELECTION OF THE DECISION-MAKING METHOD

Decision-making is a centuries-old problem; many publications have been dedicated to this topic. Hence, some assumptions must be declared before entering the state-of-the-art. Moreover, the problem described above implies a discrete nature of the alternatives. Furthermore, presumably, some criteria can be only described on an ordinal scale like a grading system. Finally, the purpose is to select the best alternatives from a given set, or to check, if a new alternative is beneficial. The complex situation will probably lead to numerous alternatives.

According to Hwang et al. in [7], the application of MCDM is widespread. However, there are some common characteristics between them: (i) incommensurable units, (ii) conflict between criteria, (iii) multiple objectives/attributes, and (iv) design/selection.

Some authors have divided MCDM into two categories. First, Multi-Attribute Decision-Making (MADM) focuses on problems with discrete decision spaces. Second, Multi-Objective Decision-Making (MODM) problems involve several competing objectives that need to be optimised simultaneously [8].

An MODM problem is associated with the problem of designing optimal solutions through mathematical programming. The number of possible decision alternatives can be immense. Usually, the decision space is continuous [9]. As common characteristics, MODM methods have: (i) a set of quantifiable objectives, (ii) a set of well-defined constraints, and (iii) a process of obtaining some trade-off information between quantifiable objectives and non-quantifiable objectives [7].

MADM requires that the choice is being made with clearly defined criteria. MADM problems have predetermined and limited number of alternatives; hence the decision space is discrete. Solving a MADM problem requires ordering and ranking [9, 10].

Comparing MODM and MADM, MADM seem to suit better the peculiarities of this paper's problem. The evaluation within a discrete decision space with predefined alternatives and criteria fits the declared assumptions. The number of alternatives is finite, although large.

Majdi divides in [11] MADM into four groups: Cost-Benefit Analysis (CBA), Elementary, Multi-Attribute Utility Theory (MAUT), and Outranking. CBA evaluates on a monetary basis the costs and benefits of the alternatives. Elementary methods do not need computation support and can be used with a few alternatives and criteria with a single decision-maker [12]. Examples of elementary methods are the Pros and Cons Analysis, the Maximin, and the Maximax Methods [11].

For the MAUT methods, Winterfeldt et al. described in [11, 13 apud] the procedure as: (i) evaluate alternatives, (ii) assign weights, (iii) aggregate the weights of attributes and alternative scores, and (iv) perform sensitivity analyses and make recommendations. For example, the Analytic Hierarchy Process (AHP) is a widely used method in this class. Advantages are the possibility to use qualitative and quantitative criteria and good traceability [14].

Outranking methods require specifying alternatives, criteria, and the use of data from the decision table. For example, the ELECTRE family (ELimination Et Choix Traduisant la REalite) consists of seven different models derived from the original one. The result is the smallest set of the best alternatives while providing no ranking with such a set [14].

The PROMETHEE approach (Preference Ranking Organisation Method for Enrichment Evaluations), described by Brans et al. in [15], is another outranking method based on extensions of the notion of criteria and can be relatively rapidly built by the decision maker. There are two base possibilities to provide rankings in this method: PROMETHEE I provide a partial pre-order, and PROMETHEE II the total pre-order. As per de Almeida et al. in [16], the method was for example extended for a range assessment in PROMETHEE III and the application on continuous decision spaces in PROMETHEE IV. According to Brans et al., PROMETHEE II is easier to handle by the decision maker. However, PROMETHEE I contain more realistic information, especially regarding incompatibilities [17].

PROMETHEE I considers the intersection between the positive and negative flows in a partial pre-order between the alternatives. The ranking of this partial pre-order can be represented as a network graph and contains information on the comparability of two alternatives. Non-comparability equals a not confirmed outrank. The combination of in- and out-flow determines if one alternative is outranking another or is indifferent.

PROMETHEE II classifies the alternatives, establishing a complete pre-order among all the alternatives using the net-flow. The alternatives with the higher net-flow are preferred over the ones with a lower net-flow.

Disadvantage of this method is that it is hard to keep an overview of the problem when many criteria are involved, and it can be time-consuming [14]. Despite those drawbacks, PROMETHEE was selected since it is widely used, does not require normalisation, and is applicable even when information is missing. Furthermore, there are many methods to assist in choosing the best option from a set of alternatives based on multiple criteria. However, it can be challenging to assess which method is the most appropriate to use in each situation or even which questions to ask when comparing various methods [18].

The original PROMETHEE algorithm described by Brans et al. in [15] is displayed on the left side of Table 1. Equation (1) represents the preference π for the criterion k of the alternative a_i over another alternative x. For the sake of simplicity a usual preference function was used here to evaluate the criteria value f. Each criterion has a weight w assigned by the decision maker. Again, for simplicity, an equal weight for all q criteria were chosen. In the first step, the preference of one alternative over all other alternatives is calculated according to equation (3), where n is the total number of alternatives A. Next, the PROMETHEE I in-flow ϕ^+ by equation (4) and out-flow ϕ^- by equation (6) is determined. Finally, the PROMETHEE II total pre-order in form of the net-flow ϕ is derived in equation (9).

Analysing the algorithm, the time complexity is $O(qn^2)$. PROMETHEE is a fully deterministic procedure; the same input will lead to the same output. However, there are some instabilities regarding the pre-order; also described, e. g. by de Keyser et al. in [19], as the reverse rank problem. While in the direct comparison of two alternatives, the preference matrix remains the same, the flow calculation introduces a dependency of the pre-order on the compared alternatives. The reverse rank problem requires the re-assessment of all alternatives if a new one is added. Revisiting the above-declared assumptions and the time complexity, PROMETHEE seems to face a significant hurdle.

Calders et al. proposed in [20] an adaption of the original algorithm achieving a time complexity of $O(qn \log n)$. This massively reduced complexity enables the computation of huge numbers of alternatives. As shown on the right side of Table 1, the uni-criterion flows are calculated according to equations (5), (7), and (8) as the initial step. Calders et al. observed that the values of a criterion per alternative can be sorted individually, allowing to infuse established and highly efficient sorting algorithms leading finally to reduced time complexity. Comparing equations (11) and (12), it becomes clear that for PROMETHEE II, the complete pre-order is for both methods the same. As a drawback, only the PROMETHEE II result can be obtained.

Table 1.
Comparison of original and sorting-based algorithms

$$\pi_k(a_i, x) = \begin{cases} 0 \text{ if } f_k(a_i) \geq f_k(x) \\ 1 \text{ if } f_k(a_i) < f_k(x) \end{cases} \qquad (1)$$

$$w_k = {}^1\!/_q \qquad (2)$$

Original PROMETHEE I & II [15]	Sorting Based PROMETHEE II [20]
$\pi(a_i, x) = \sum_{k=1}^{q} [w_k * \pi_k(a_i, x)]$ (3)	
$\phi^+(a_i) = \frac{1}{n-1} \sum_{x \in A} \pi(a_i, x)$ (4)	$\phi_k^+(a_i) = \frac{1}{n-1} \sum_{x \in A} \pi_k(a_i, x)$ (5)
$\phi^-(a_i) = \frac{1}{n-1} \sum_{x \in A} \pi(x, a_i)$ (6)	$\phi_k^-(a_i) = \frac{1}{n-1} \sum_{x \in A} \pi_k(x, a_i)$ (7)
	$\phi_k(a_i) = \phi_k^+(a_i) - \phi_k^-(a_i)$ (8)
$\phi(a_i) = \phi^+(a_i) - \phi^-(a_i)$ (9)	$\phi(a_i) = \sum_{k=1}^{q} [w_k * \phi_k(a_i)]$ (10)
$\phi(a_i) = \frac{1}{q(n-1)} \sum_{x \in A} \left[\sum_{k=1}^{q} [\pi_k(a_i, x) - \pi_k(x, a_i)] \right]$ (11)	$\phi(a_i) = \frac{1}{q(n-1)} \sum_{k=1}^{q} \left[\sum_{x \in A} [\pi_k(a_i, x) - \pi_k(x, a_i)] \right]$ (12)

PROPOSED METHOD

In this paper a stepwise method to the MCDM problem is proposed. As displayed in Figure 1, the approach consists of the definition of an initial criteria list, the derivation of a final criteria list, and the decision making.

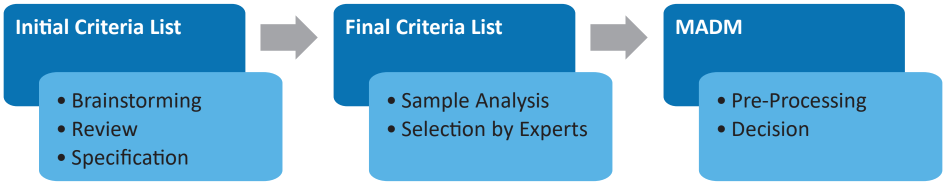


Figure 1. Proposed method flow

Initial Criteria List

Brainstorming phase To define criteria, a brainstorming session with experts is proposed. Categories of cost and use factors can support collecting the criteria.

The central use of a metamodel is determined by its estimation quality. This quality should not only be defined by typical metrics such as accuracy or recall but also consider the detail of degree and the relevance of an estimation. Furthermore, the difference in result generation between the assessed alternative and the simple FEA simulation can be considered.

The main cost factor is induced by the data for training and assessment of the metamodel. If the architecture requires additional data as input for each estimation, it adds to the costs. Depending on the data volume, the computational cost for the training and assessment cycle and even per prediction can be relevant.

The end of a model's validity should be considered. In this case, additional costs through data generation for retraining the metamodel will occur. Furthermore, it is assumed that over time and continuous development, the vehicle deviates increasingly from the ones used for metamodel training. Hence, a later loss of validity – or a wider validity range – would mean a higher model value.

Review & specification phase Typically, brainstorming techniques are suitable for collecting ideas efficiently; however, completeness is not guaranteed. Hence, a review checking the inner logic and completeness of the criteria is recommendable. Dropping criteria in this step is unnecessary; this will be done in the final specification phase.

During the review phase, the reporting scale and assessment method should be defined for each criterion. The reporting scale will influence the selection of a suitable MADM method. The assessment method's exact definition will help to review the selected criteria and is the prerequisite for the later steps. Since criteria for the actual assessment should be selected later, the documentation of each criterion and its motivation is necessary.

Derivation of the Final Criteria List

Ideally, the list of criteria from the above steps can assess all relevant aspects of possible alternatives. However, highly correlated criteria are likely to occur since the described approach prefers adding criteria over dropping them. Therefore, the authors propose to create multiple samples of alternatives. Those should be used to test the validity and plausibility of the defined assessment algorithms and for another review phase. The samples can support the identification of highly correlated criteria. Those criteria would potentially assess the same aspect; hence assign a higher weight to such an aspect.

It must be noted that the sample alternatives will not cover all possible cases. Henceforth, expert opinion is needed to interpret the findings correctly. Each criterion and the related findings should be discussed considering the aspects described in Table 2.

Table 2.
Aspects to consider during criterion-selection

Representativeness
The representativeness of the generated samples determines if the criteria are correct and meaningful and represent diverse aspects of the problem.
Correlation
Correlated criteria should be merged to avoid unwanted higher weights on a specific aspect. Invariant criteria should be inspected if the invariance is only due to the generated samples or meaningful for the overall problem. In the first case, the criterion can remain, in the ladder, dropped.
Transparency & Directness
The criteria should be grouped into meaningful categories to support a transparent rating scheme. It depends on the actual use case to which category an aggregated criterium fits. Another aspect regarding transparency is the understandability of a criterium. A directly assessed criterium is more straightforward to process and understand than one resulting from complex calculations.

Level of the scales

Ultimately, the reporting scale of a criterion should be considered. Typically, nominal-, ordinal-, interval-, and ratio-scales are used, where nominal has the lowest and ratio the highest level. The lowest-level scale of all used ones will determine which MCDM method can be utilised. Not all criteria can be assessed on ratio scales. However, if possible, the higher-level scale always seems preferable.

Multi-Attribute Decision Making

Following the state-of-the-art analysis, to solve the decision-making problem, the PROMETHEE II method was selected. The adaption of Calders et al. in [20] seems recommendable to deal with the expected high number of alternatives despite the loss of the PROMETHEE I result.

The method required selecting a preference function; it is influenced by the lowest scale order and the user's taste. If of all criteria, the lowest order scale is ordinal, only the usual-criterion can be used. Other definitions, like the linear- or step-criterion, require proportional intervals between the variables.

APPLICATION

Computations were executed on a workstation equipped with an Intel Xeon W-2123 CPU with 3.6 GHz and 64 GB RAM. The cluster used for the larger FE-simulations had per node two Intel Xeon E5-2687W v4 CPUs with 3 GHz.

All described algorithms were implemented in Python 3.8. The neural network used for machine learning was taken from the Scikit-Learn library [21] version 1.0.1. Database-related operations like sorting were done utilising the Pandas library [22] version 1.4.2. All FE-simulations were performed in LS-Dyna 10.0 (MPP on cluster, SMP on workstation) with single precision.

Database

FE-model To test the method assessment approach, a database from a recent study [6] was used. The simulations were done with a 2D rigid body model, as shown in Figure 2, representing an occupant undergoing a frontal crash. Five anthropometrical configurations were created, orienting on the common crash test dummies with the 5th, 50th, and 95th percentiles. The 25th and 75th percentiles were added by interpolation. A Full Factorial Design of Experiment (DoE) was defined, containing the variation of backrest angle, seat ramp angle, impact speed, and the force of the shoulder belt load limiter. Each factor was varied in six levels, and the resulting DOE of size 1,296 was repeated for the five occupant sizes leading to a total of 6,480 simulations.

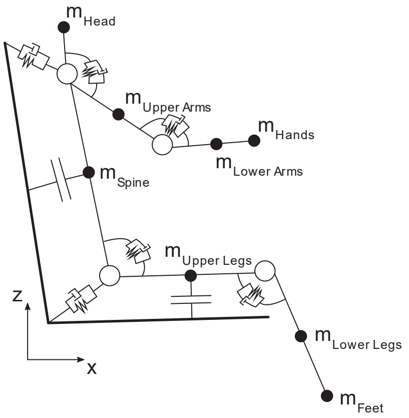


Figure 2. Occupant model 2D

The model has not been validated; thus, the physical behaviour seems overall plausible. Few simulations suffered numerical instabilities and were dropped as outliers. Following the recent study, the maximum resultant chest acceleration lasting at least 3 ms $a_{Chest,a3ms}$ and the maximum head forward displacement relative to the vehicle $x_{Head,Local}$ was selected for the investigations. The selection was motivated by stable numerical outputs, the model's capabilities, and biomechanical relevance.

For the seat, a motion was prescribed, taking the pulse generated by FE-simulation with a Toyota Yaris 2010 model [23]. The load cases were defined as vehicles crashing frontally into a rigid barrier with different velocities. One crash simulation took approximately three hours using a single node on the cluster. In comparison, one occupant simulation on the workstation accounted for approximately three minutes.

Machine learning architecture For the machine learning, in a previous study [6] an architecture was sketched as depicted in Figure 3. The key characteristic of this architecture is the hybrid approach of providing a calibration simulation for each prediction. The calibration is a physical simulation of an anthropometrical reference configuration. The predicted outputs of the metamodel are selected results of different anthropometrical configurations, but in the same vehicle environment as the reference. As learning algorithm, a deep neural network with two hidden layers was selected.

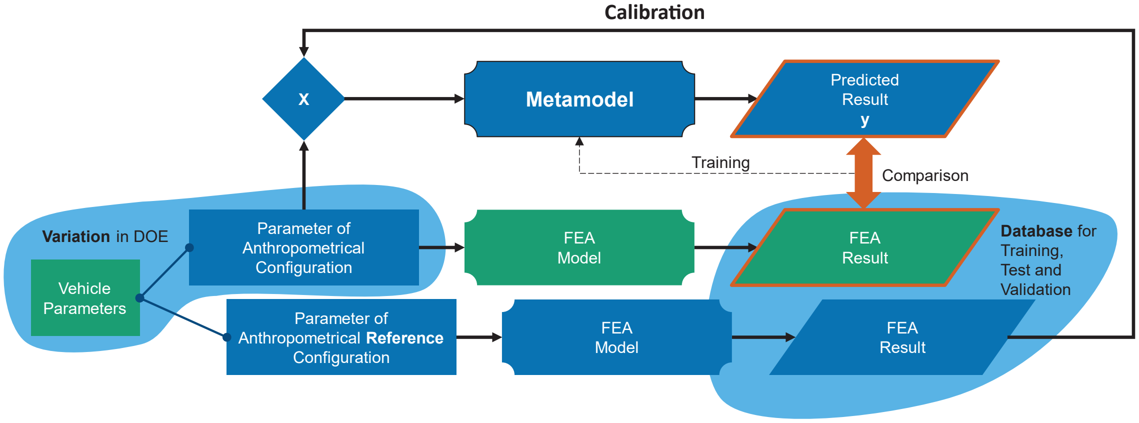

Calibration
x
Metamodel
Predicted Result y
Training
Comparison
Variation in DOE
Vehicle Parameters
Parameter of Anthropometrical Configuration
FEA Model
FEA Result
Database for Training, Test and Validation
Parameter of Anthropometrical Reference Configuration
FEA Model
FEA Result

Figure 3. Machine learning architecture in [6]

For later assessment, a characterisation of the environment must be defined. Vehicle parameters are not explicitly given but implicitly contained in the calibration simulation. The advantage is that changes in vehicles that are different than expected or unquantifiable can be covered. The drawback, however, is an unclear defined field of variation in which the metamodel was trained, the so-called validity field. Hence, the transition between interpolation and extrapolation cannot be derived directly. However, the calibration simulation contains information on the environment since a unique setting will result in a unique response.

In the case of the used 2D model, the occupant's behaviour can be described by the kinematics of the joint and endpoints (head, shoulder, elbow, hand, hip, knee, foot). The relative displacement to the vehicle of those points and the global acceleration provides sufficient insight. To measure the position of an environment relative to the validity field in a transparent manner, a reduction to one or two dimensions seems necessary.

Principal Component Analysis (PCA) was selected, and its Scikit-Learn implementation was used. This linear and self-centring method derives from high dimensional data principal components by eigenvalue decomposition, explaining the variance in the data. Such components do not necessarily correspond with single DOE parameters; they can be combinations of them, too.

Before applying PCA, the data had to be transformed. First, the sensor signals were smoothed using a CFC60 filter [24], and simulations suffering numerical instabilities were removed. Second, the sensor output time series were arranged line-wise. Each column was a discrete timestamp from a particular sensor as a dimension. Third, each line contains the data from one FE-simulation as samples. Last, each dimension/column was standardised by subtracting the dimension's mean value from each sample and then dividing it by the standard deviation.

Applied to the dataset of 50th percentiles, as displayed in Figure 4, the first principal component explains ca. 38 % of the variance and the second additionally ca. 14 %. A 6x6 field of distinct islands is observable. The first component could be associated with the six discrete impact speed settings, and the second one with the six discrete backrest angle settings from the DOE. Despite the relatively low explained variance of the principal components, only the first one was used for further processing since it could have been associated with the impact velocity and for simplicity in showcasing the actual decision-making method.

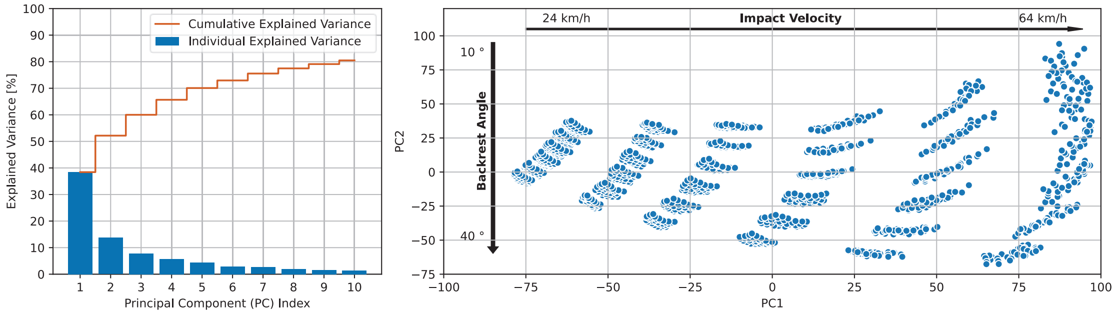


Figure 4. Result of PCA analysis

Due to the dimensional reduction, it was possible to differentiate between interpolation and extrapolation. Therefore, data from the interpolation field was used for training and testing the metamodel. The data from the extrapolation field was used for validation.

Generated alternatives In combination with the given database of physical models, the selected architecture allows to investigate numerous alternatives to create a metamodel. First, the calibration was varied between the 50th percentile and the other edge percentiles. Second, the calibration contained $a_{\mathrm{Chest,a3ms}}$, $x_{\mathrm{Head,Local}}$, or both. Third, the same variation was used for the predictions. Those labels can be defined in two or more classes or as continuous values. The number of predictable percentiles is directly dependent on the used calibrators. A percentile used as a calibrator cannot be utilised in the predictions. Finally, the hyperparameters of the neural network (number of layers, number of neurons per layer) were varied. The first component of the PCA on the 50th percentile data was selected characterise to the environment. Through this, the interpolation field could be varied as well as the number of simulations in it. In total, 12,960 alternatives were generated.

Initial List of Criteria

Brainstorming phase Following the method described above, the three categories, metamodel-setup-cost, usage, and validity-range, were defined in the first step. Those categories represent the live cycle cost and use factors. Next, a group of experts filled the categories with relevant criteria in a brainstorming session.

Criteria in the metamodel-setup-cost category should assess all occurring costs associated with creating a metamodel. Therefore, the category was differentiated into the cost for the physical simulation database, the training costs, the testing, and the assessment of the validity range.

The usage category focuses on the live cycle phase in which the metamodel is utilised. In this phase, costs for each prediction exist, but the value of the predictions is also shown.

For the last category, the validity range, it is assumed that at one point, the vehicle under development deviates so much from the ones used for the metamodel setup that its validity is compromised. In this case, costs for retraining or tuning will occur.

Review & specification phase After the brainstorming session, the experts reviewed and restructured the criteria and defined their reporting scales. The selected criteria are discussed below and are listed in Table 4 of the appendix. As documented in the table, ultimately, not all criteria were found to be implementable.

As metamodel-setup-costs, criteria assessing computation time and the number of simulations or samples were accounted. It was differentiated between computation time for the crash and occupant simulations (~ 3 h, ~ 3 min). Computation time reports on a continuous scale whereas the sample number on a discrete scale. For both, lower is seen as better.

In the usage section, the use and value of a metamodel were locked from several angles. First is the value from the prediction type; a binary classification is seen to have a lower value than a continuous regression. A rating system was used as a metric. Second, a single sensor's output detail can determine the value. With decreasing value, the prediction of the entire sensor output as time series, the prediction of relevant output characteristics, and finally, the prediction of a single value was defined in a rating system. Third, a crash test dummy is instrumented with numerous sensors. More the sensors are used, higher the value. This criterion was defined as the number of not used sensors to fit into the lower-is-better scheme. Of course, not all sensors have the same relevance. The defined criterion reports by relevant legislation, consumer ratings, and physics in a ranking scale. Finally, the granularity of the predicted anthropometrical configuration can range from a single configuration over distinct percentiles up to the variation of anthropometrical measures.

The most apparent and commonly used criterion is the prediction quality metric. For regression, the coefficient of determination R^2 was used. For classification cases, the F-score was selected. Both metrics report to a continuous scale where one is the best. The R^2-score can take negative values; to adapt its scale to F-score, equation (13) was defined.

$$R^2 = \begin{cases} 0 \text{ if } R^2 < 0 \\ R^2 \text{ if } R^2 \geq 0 \end{cases} \qquad (13)$$

As described above, the environment was characterised as 1D through PCA. As the value of a metamodel increases, a new environment can differ from more the training field without compromising the model's validity. The machine learning metric was evaluated, as displayed in Figure 5, for the inter- and extrapolation zones separately to assess the width. Each zone was split into three segments to get a gradual result. It must be noted that a machine learning metric is a statistical measure and hence, needs an appropriate sample size to deliver a valid assessment [25]. Finally, the width results from the area in which the machine learning metric is continuously higher than 0.8. The assessed machine learning score was defined as the mean value of the machine learning metric over that width. Again, the machine learning metric was subtracted from one to achieve the lower-is-better scale, and the width was multiplied by minus one.

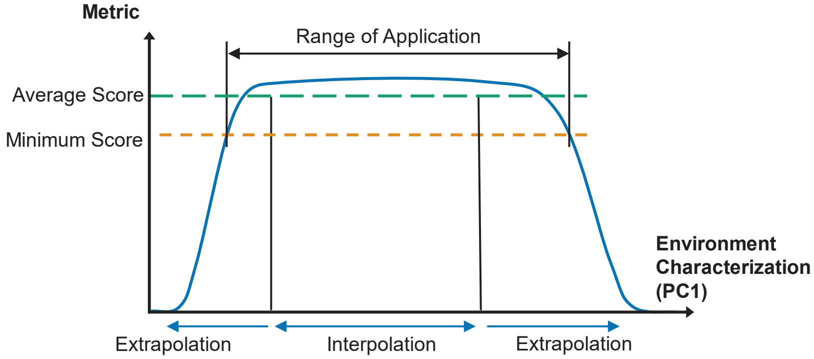


Figure 5. Concept of interpolation, extrapolation, and validity range

Final List of Criteria

To review the listed criteria and filter them, they were assessed in the 12,960 sample alternatives. For statistical insight, the Kendall correlation coefficient was used. This coefficient considers the rank correlation between two criteria and can deal with limited non-linearity and outliers.

The purposes of the correlation analysis were to support (i) the identification of double-assessed properties and (ii) the reasonability of the assessments. A few mistakes and misalignments in the assessment algorithms could be identified during this process. Furthermore, some criteria were found to be correlated or invariant.

Invariance of criteria occurred since the sample alternatives did not cover all possibilities identified by the experts. Such criteria were kept. Highly correlated criteria were merged.

Criteria related to computation time were under discussion. The values used for the computation time of the FE-models for crash and occupant simulation were average values; hence reliable. In contrast, the times from the instrumented assessment codes were measured only once. Hence, disturbances on the CPU, e. g. other processes, can lead to incomparable times for computation. However, for this paper, it was possible to run the process on a CPU exclusively. A statistical univariant analysis showed no extreme outliers; an example is shown in Figure 6. On this base, it was decided to keep those criteria since no adequate alternative could be identified.

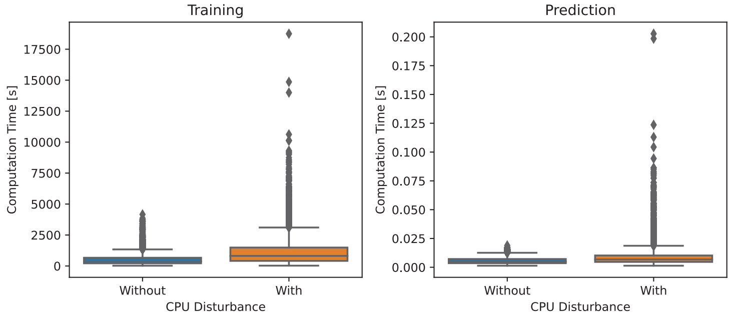


Figure 6. Computation time distribution with and without disturbances

Ultimately, 28 criteria were selected for further usage. During the selection process, 15 criteria were dropped. Also, the not implementable ones were removed. The final list is provided in Table 5 of the appendix.

Multi-Attribute Decision Making

Sorting method The alternation of the original PROMETHEE algorithm proposed by Calders et al. was implemented. The sorting-based method was described for a maximise-problem and a linear preference criterion. In comparison, the here used implementation inverted the comparing algorithm to a minimise-problem. The lowest order scale type in the final list of criteria was the ordinal scale. Hence, only the usual criterion could be used. The algorithm was adapted accordingly. For sorting, the MERGESORT algorithm was used. The final algorithm, as implemented, is displayed in Table 3.

The 12,960 sample alternatives with 28 criteria were divided into 40 chunks to check the algorithm and compare the computation time. Each chunk was assessed by implementations of the original PROMETHEE algorithm and the one with the Calders modification. Running on a workstation as a single CPU process, the median computation time of the original algorithm was 80.1 s and of the sorting method 0.1 s. The PROMETHEE II net-flows were within the limits of the computational precision same. The equality of both approaches and the drastically lower time complexity of the sorted approach was confirmed.

Table 3.
Algorithmic representation of PROMETHEE II implementation

```
Input: Number of alternatives $n$, Number of criteria $q$, criteria values assessed for all alternatives $f_{1...q}(a_{1...n})$
Return: Net-flows $\phi(a_{1...n})$

SET $w$ TO $^1/_q$                                        # Equal weight per criterion
INIT $\phi_{1...q}[a_{1...n}]$                             # Uni-criterion net-flow
FOR $k$ IN $f[a_0]$
    INIT $\phi_k^{+/-}[a_{1...n}]$                         # Uni-criterion in- / out-flow
    FOR $\mathbb{S}$ IN [1, -1]:                           # In- & outflows
        SET $\mathbb{f}_k(a)$ TO $f_k(a) * \mathbb{S}$     # Symmetry of flows
        SET $\mathbb{f}_k(\mathbb{a})$ TO SORTED_DESCENDING $\mathbb{f}_k(a)$   # Merge Sort ($a$ differs from $\mathbb{a}$ only in its order)
        SET $R$ TO $\mathbb{f}_k(\mathbb{a})$              # For the first object, all others are on the right
        SET $\phi_k^{\mathbb{S}}[\mathbb{a}_1]$ TO 0       # The first alternative in order always has flow 0
        FOR $i$ IN $\mathbb{a}_{2...n}$:                   # Loop over the following alternatives in order
            SET $\phi_k^{\mathbb{S}}[\mathbb{a}_i]$ TO $\phi_k^{\mathbb{S}}[\mathbb{a}_{i-1}]$   # Start with the previous flow
            WHILE $R[\mathbb{a}_1] > \mathbb{f}_k[\mathbb{a}_i]$   # Check for preference
                DELETE $R[\mathbb{a}_1]$                   # Move to left
                SET $\phi_k^{\mathbb{S}}[\mathbb{a}_{i-1}]$ TO $\phi_k^{\mathbb{S}}[\mathbb{a}_{i-1}] + \frac{1}{n+1}$   # Add as the preference to uni-criterion in- / out-flow
    FOR $i$ IN $a_{1...n}$
        SET $\phi_k(a_i)$ TO $\phi_k^+(a_i) - \phi_k^-(a_i)$   # Uni-criterion net-flow
INIT $\phi(a_{1...n})$                                     # Net-flow
FOR $i$ IN $a_{1...n}$
    SET $\phi(a_i)$ TO $\sum_{k=1}^{q}(w * \phi_k(a_i))$   # Net-flow
```

<u>Result</u> For the final assessment, all alternatives were tested as a whole and sorted by their PROMETHEE II complete pre-order. As defined above, seven parameters were varied: (i) the configuration of the neural network, (ii) prediction type, (iii) target percentile(s), (iv) calibrating percentile(s), (v) sample size, (vi) interpolation range, (vii) sensor(s) used in target(s), and (viii) sensor(s) used in feature(s). The tendencies, observed in Figure 7, are described below.

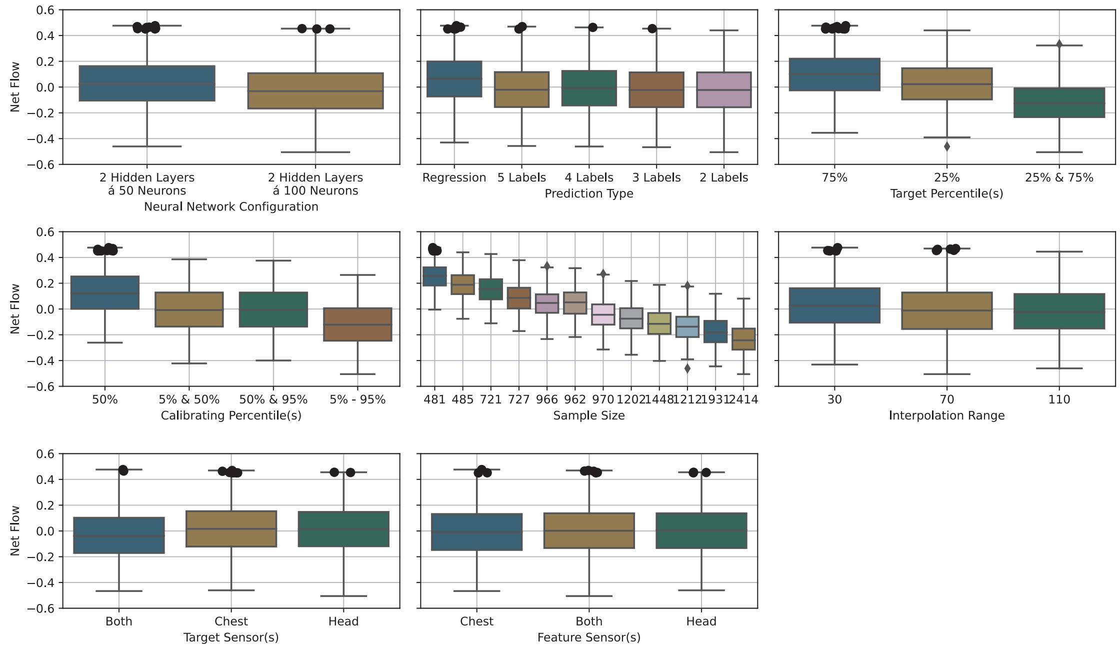


Figure 7. Net-flows evaluated for 12,459 alternatives – box plots with top 10 overlay

The results indicated a negative influence of sample size on the rank. One reason can be the increased cost of data generation, while other factors overlay the potential positive influence on the prediction quality.

Furthermore, the regression algorithms seemed slightly better than the others. The 95th or 5th percentile prediction seemed to be more successful than simultaneously targeting both. Predicting the 95th percentile is indicated as beneficial. Using the 50th percentile as a calibrator seems to be better than the 5th or 95th percentile. Taking the 5th and 95th percentile as calibrators does not seem advantageous. Finally, it seems that a tighter interpolation field has light benefits. The other varied parameters do not indicate preferences.

Concluding, the alternatives could be ranked using PROMETHEE II. The first analysed tendencies seem to be reasonable. In general, settings which compromise the prediction value have a strong influence.

DISCUSSION

The process of deriving the list of criteria seems, overall, a good concept. Nevertheless, the strong dependency of all steps on the knowledge and judgement of the involved experts must be pointed out.

The results of a brainstorming process can be unorganised, and there is no guarantee of completeness. Additionally, there is a chance for non-implementable criteria. The pre-declaration of some categories representing the main cost and use factors was very helpful. It is highly recommendable to invest already during that first phase in documenting each criterion's intentions. In the last review step, each criterion should be described extensively, helping to keep the overview and to succeed in the later steps. Concluding, with the proposed process, a comprehensive list can be created.

For the sake of simplicity, it seems recommendable to define all scales in a lower is a better manner. However, this is not required by algorithms as PROMETHEE. Furthermore, the ordinal scale can be applied to all criteria and can be assessed transparently. However, the choice of this scale limits the usable decision-making methods. Already using another preference function within PROMETHEE would transform the scale unwanted and unreasonably into a ratio scale [19].

As stated above, a set of test samples cannot represent all possible variations. The high number of alternatives used in this paper was mainly motivated to ensure a good range of variation and to enable the investigation of correlations. In the end, the correlations did not lead to a data-driven decision over the criteria. However, as a tool to detect unplausible behaviour, it was invaluable. It can be achieved with a significantly smaller number of test alternatives. By experts' judgement, a minimum number covering a maximal range of variations can be defined.

The results from the ranked alternatives originate in the 2D FE-model. A significant limitation can be found in the characterisation of the environment. First, the explained variance seemed insufficient even if the dimensional reduction showed a physically relatable result. In future studies, a detailed analysis on the base of a validated FE-model should be conducted and the method for dimensional reduction refined. Second, especially the criteria were defined for the narrow use case of supervised machine learning for the virtual assessment of occupant crash safety. If the method should be applied in deviating domains, each step starting with the declaration of the initial categories, should be reviewed. Depending on the complexity, increasing the number of test samples seems apt.

If changing the MADM method, the investigations on its behaviour and the parametric sensitivity should be done. Furthermore, especially if the exact rank of the assessed alternatives is relevant, the rank reversal issue of pairwise comparison-based methods, in general, but especially PROMETHEE II, should be assessed.

CONCLUSIONS AND OUTLOOK

The selection of an appropriate setup of a machine learning architecture and its pipeline was framed as a multi-attribute discussion-making problem. The proposed method was developed for a rapid occupant safety assessment with a particular supervised learning setup.

The proposed method consists of the decision-making preparation containing (i) the definition of an initial list of criteria and (ii) the review of them using sample alternatives, leading to (iii) the definition of the final criteria list. From the literature research, the PROMETHEE II decision-making method was selected. A version of the sorting-based algorithm proposed by Calders et al. was implemented.

The method was tested on data from a finite element model in the validation part. A final list of criteria was developed and used to rank sample alternatives resulting from a parameter variation. First tendencies of the influence of the alternative's parameters on its rank could be identified.

The method was discussed, and recommendations were derived. Overall, a high dependency on expert knowledge was identified. For the criteria, ordinal scales seemed apt. PROMETHEE II, with the sorting algorithm, delivered a plausible and distinct ranking, and the time complexity allowed the assessment of an immense number of alternatives simultaneously.

The method should be applied to a database based on a more realistic and validated finite element model. Further research will be dedicated to the vehicle characterisation for more than one dimension and to the dimensional reduction approach. The increasing need for efficient assessment methods will fuel further validation.

REFERENCES

[1] Ratingen, M.R. van, “Euro NCAP – from passive safety to assistance systems and beyond,” *crash.tech 2022*, Ingolstadt, 2022.

[2] Östling, M. and Larsson, A., “Occupant Activities and Sitting Positions in Automated Vehicles in China and Sweden,” *26th ESV*, Eindhoven, Netherlands, 2019.

[3] Reed, M.P. and Rupp, J.D., “An anthropometric comparison of current ATDs with the U.S. adult population,” *Traffic Injury Prevention* 14(7):703–705, 2013, doi:10.1080/15389588.2012.752819.

[4] Wang, S.C., Hsieh, C.-H., Cheng, C.-T., Chiu, C.-H. et al., “Morphometric Characterisation of an Asian Reference Analytic Morphomics Population (A-RAMP),” *IRCOBI Conference*, Florence, Italy, 2019.

[5] Plaschkies, F., Vaculin, O., and Schumacher, A., “Assessment of the Influence of Human Body Diversity on Passive Safety Systems: A State-of-the-art Overview,” *FISITA Web Congress*, doi:10.46720/F2021-PIF-071, 2021.

[6] Plaschkies, F., Vaculin, O., Pelisson, A., and Schumacher, A., “Schnelle Abschätzung des Crashverhaltens von Insassen unter Berücksichtigung der Vielfalt des Menschen: Robustheit, Datenintensität und Vorhersagekraft von Metamodellen,” *VDI Fahrzeugsicherheit*, Berlin, doi:10.51202/9783181023877-313, 2022.

[7] Hwang, C.-L. and Yoon, K., “Multiple Attribute Decision Making: Methods and Applications A State-of-the-Art Survey,” Springer eBook Collection, vol. 186, Springer, Berlin, Heidelberg, ISBN 978-3-642-48318-9, 1981.

[8] Zavadskas, E.K., Antucheviciene, J., and Kar, S., “Multi-Objective and Multi-Attribute Optimization for Sustainable Development Decision Aiding,” *Sustainability* 11(11):3069, 2019, doi:10.3390/su11113069.

[9] Kahraman, C., “Fuzzy Multi-Criteria Decision Making,” vol. 16, Springer US, Boston, MA, ISBN 978-0-387-76812-0, 2008.

[10] Wolny, M., “Analysis of the Multiple Attribute Decision Making Problem with Incomplete Information about Preferences among the Criteria,” *MCDM* 11:187–197, 2016, doi:10.22367/mcdm.2016.11.12.

[11] Majdi, I., “Comparative evaluation of PROMETHEE and ELECTRE with application to sustainability assessment,” Master Thesis, Concordia University, Montreal, Quebec, Canada, 2013.

[12] Linkov, I., Varghese, A., Jamil, S., Seager, T.P. et al., “Multi-Criteria Decision Analysis: A Framework for Structuring Remedial Decisions at Contaminated Sites,” in: Linkov, I. and Ramadan, A.B. (eds.), *Comparative Risk Assessment and Environmental Decision Making*, Nato Science Series: IV: Earth and Environmental Sciences, Kluwer Academic Publishers, Dordrecht, ISBN 1-4020-1895-9:15–54, 2005.

[13] Winterfeldt, D. von and Edwards, W., “Decision analysis and behavioral research,” Univ. Pr, Cambridge, ISBN 978-0521273046, 1986.

[14] Ayağ, Z., “An approach to evaluate CAM software alternatives,” *International Journal of Computer Integrated Manufacturing* 33(5):504–514, 2020, doi:10.1080/0951192X.2020.1757156.

[15] Brans, J.P. and Vincke, P., “Note—A Preference Ranking Organisation Method,” *Management Science* 31(6):647–656, 1985, doi:10.1287/mnsc.31.6.647.

[16] Almeida, A.T. de and Costa, A.P.C.S., “Modelo de decisão multicritério para priorização de sistemas de informação com base no método PROMETHEE,” *Gest. Prod.* 9(2):201–214, 2002, doi:10.1590/S0104-530X2002000200007.

[17] Brans, J.P., Vincke, P., and Mareschal, B., “How to select and how to rank projects: The Promethee method,” *European Journal of Operational Research* 24(2):228–238, 1986, doi:10.1016/0377-2217(86)90044-5.

[18] Akhavi, F. and Hayes, C., “A comparison of two multi-criteria decision-making techniques,” *SMC'03 Conference Proceedings.*, Washington, DC, USA, IEEE, doi:10.1109/ICSMC.2003.1243938, ISBN 0-7803-7952-7:956–961, 2003.

[19] Keyser, W. de and Peeters, P., “A note on the use of PROMETHEE multicriteria methods,” *European Journal of Operational Research* 89(3):457–461, 1996, doi:10.1016/0377-2217(94)00307-6.

[20] Calders, T. and Assche, D. van, “PROMETHEE is not quadratic: An O(qnlog(n)) algorithm,” *Omega* 76:63–69, 2018, doi:10.1016/j.omega.2017.04.003.

[21] Pedregosa, F., Varoquaux, G., Gramfort, A., Verleysen, M. et al., “Scikit-learn: Machine Learning in Python,” *Journal of Machine Learning Research* 12:2825–2830, 2011.

[22] The pandas development team, pandas-dev/pandas: Pandas, Zenodo, 2022.

[23] Marzougui, D., Samaha, R.R., Cui, C., Kan, C.-D. et al., “Extended Validation of the Finite Element Model for the 2010 Toyota Yaris Passenger Sedan,” NCAC 2012-W-005, 2012.

[24] “Instrumentation for Impact Test - Part 1 - Electronic Instrumentation: SAE J211/1,” in: *SURFACE VEHICLE RECOMMENDED PRACTICE*, 2014.

[25] Powers, D.M.W., “Evaluation: From Precision, Recall and F-Factor to ROC, Informedness, Markedness & Correlation,” *Journal of Machine Learning Technologies* 2(1):37–63, 2011.

ACKNOWLEDGEMENT

The authors acknowledge the kind support of the German Academic Exchange Service (DAAD) in the project enGlobe, which supported the research in the frame of the Bavarian Center for Applied Research and technology with Latin America (AWARE).

APPENDIX

Table 4.
Initial list of criteria

Metamodel Setup Cost			
	Database Setup		
1	setup_db_num_sim_calibration_pprediction	Calibration simulations per environment (= per prediction)	
2	setup_db_num_sim_calibration_sum	Total number of calibration simulations in database	
3	setup_db_num_sim_crash	Total number of crash simulations in database	
4	setup_db_num_sim_occupant	Total number of occupant simulations in database	
5	setup_db_num_sim_assessment	Total number of samples used for metamodel assessment	
6	setup_db_num_sim_training	Total number of samples used for metamodel training	
7	setup_db_comp_time_calibration	Computation time for calibration simulations per prediction	
8	setup_db_comp_time_crash	Total computation time of all crash simulations in database	
9	setup_db_comp_time_occupant	Total computation time of all occupant simulations in database	
10	setup_db_comp_time_assessment	Total computation time of for assessment used samples	
	Training Phase		
11	setup_training_comp_time_metamodel	Computation time of metamodel’s training	
12	setup_training_comp_time_assessment_sum	Computation time of metamodel’s assessment on training data	
13	setup_training_comp_time_assessment_pprediction	Computation time for single prediction by metamodel	
14	setup_training_comp_time_calibration	Total computation time for calibration simulations used in training	
15	setup_training_calibration_sum	Number of calibration simulations used for training	
16	setup_training_calibration_pprediction	Calibration simulations per environment (= per prediction)	
17	setup_training_num_sim_crash	Total number of crash simulations used for training	
18	setup_training_num_sim_occupant	Total number of occupant simulations used for training	
19	setup_training_num_sim_assessment	Number of samples used for assessing in training phase (equals number of samples for training)	N/A
20	setup_training_comp_time_crash	Total computation time for crash simulations used for training	
21	setup_training_comp_time_occupant	Total computation time for occupant simulations used for training	
22	setup_training_comp_time_assessment	Total computation time of all for assessment used samples during training (equals computation time of simulations for training)	N/A
	Interpolation Assessment Phase		
23	setup_test_comp_time_assessment_sum	Computation time of metamodel’s assessment (predictions)	
24	setup_test_comp_time_assessment_pprediction	Computation time for single prediction by metamodel	
25	setup_test_comp_time_calibration	Total computation time for calibration simulations used for assessment	
26	setup_test_calibration_sum	Number of calibration simulations used for assessment	
27	setup_test_calibration_pprediction	Calibration simulations per environment	
28	setup_test_num_sim_crash	Total number of crash simulations used for assessment	
29	setup_test_num_sim_occupant	Total number of occupant simulations used for assessment	
30	setup_test_num_sim_assessment	Total number of for assessment used samples	
31	setup_test_comp_time_crash	Total computation time for crash simulations used for assessment	
32	setup_test_comp_time_occupant	Total computation time for occupant simulations used for assessment	
33	setup_test_comp_time_assessment	Total computation time of all for assessment used samples (occupant & crash)	N/A
	Validity (Extrapolation) Assessment		
34	setup_val_comp_time_assessment_sum	Total computation time for predictions in extrapolation range	
35	setup_val_comp_time_assessment_pprediction	Computation time for single prediction by metamodel	
36	setup_val_comp_time_calibration	Total computation time for calibration simulations used for assessment	N/A
37	setup_val_calibration_sum	Number of calibration simulations used for assessment	N/A
38	setup_val_calibration_pprediction	Calibration simulations per environment	N/A
39	setup_val_num_sim_crash	Total number of crash simulations used for assessment	N/A
40	setup_val_num_sim_occupant	Total number of occupant simulations used for assessment	N/A
41	setup_val_num_sim_assessment	Total number of for assessment used samples	N/A
42	setup_val_comp_time_crash	Total computation time for crash simulations used for assessment	N/A
43	setup_val_comp_time_occupant	Total computation time for occupant simulations used for assessment	N/A
44	setup_val_comp_time_assessment	Total computation time of all for assessment used samples (occupant & crash)	N/A
Usage			
45	us_metamodel_num_sim_calibration	Calibration simulations per environment	
46	us_metamodel_time_sim_calibration	Computation time for calibration simulations per environment	

47	us_metamodel_time_prediction	Computation time for single prediction by metamodel	
48	us_prediction_type	Value of prediction type (binary classification to regression)	
49	us_prediction_outputs	Degree of detail of predictions (single value to full sensor time series)	
50	us_prediction_sensor_num	Number of not used sensors of available sensors in dummy	
51	us_prediction_sensor_relevance	Relevance of used sensors (irrelevant to utilized in legislation)	
52	us_prediction_anthropometrics	Detail of degree of anthropometrical distinction	
53	us_MLmetric	Value from assessed metric	
Validity Range			
54	val_range	Width of validity range	
55	val_range_retraining	Cost to retrain the metamodel	N/A

N/A – not implemented

Table 5
List of final criteria

	Name	**Description**	
1	setup_db_num_sim_crash	Total number of crash simulations in database	
2	setup_db_comp_time_crash	Total computation time of all crash simulations in database	
3	setup_db_num_sim_assessment	Total number of samples used for metamodel assessment (test & validation)	
4	setup_db_num_sim_training	Total number of samples used for metamodel training	
5	setup_training_calibration_sum	Number of calibration simulations used for training	
6	setup_training_comp_time_assessment_pprediction	Computation time for single prediction by metamodel	
7	setup_training_comp_time_assessment_sum	Computation time of metamodel’s assessment (predictions) on training data	
8	setup_training_comp_time_calibration	Total computation time for calibration simulations used in training	
9	setup_training_comp_time_crash	Total computation time for crash simulations used for training	
10	setup_training_comp_time_metamodel	Computation time of metamodel’s training	
11	setup_training_num_sim_crash	Total number of crash simulations used for training	
12	setup_training_num_sim_occupant	Total number of occupant simulations used for training	
13	setup_test_comp_time_crash	Total computation time of all crash simulations in database	
14	setup_test_num_sim_crash	Total number of crash simulations used for assessment	
15	setup_test_comp_time_assessment_sum	Computation time of metamodel’s assessment (predictions)	
16	setup_test_calibration_sum	Number of calibration simulations used for assessment	
17	setup_test_num_sim_occupant	Total number of occupant simulations used for assessment	
18	setup_test_comp_time_assessment_pprediction	Computation time for single prediction by metamodel in test phase	
19	setup_val_comp_time_assessment_pprediction	Computation time for single prediction by metamodel in validation phase	
20	setup_val_comp_time_assessment_sum	Total computation time for predictions in extrapolation range	
21	us_Mlmetric	Value from assessed metric; F-score for classification / R²-score for regression	
22	us_prediction_anthropometrics	Detail of degree of anthropometrical prediction, grades, where 1 is best	
		5	1 percentile
		4	2 percentiles
		3	3-4 percentiles
		2	≥ 5 percentiles
		1	Anthropometrical parameter
23	us_prediction_outputs	Degree of detail of predictions, grades, where 1 is best	
		3	Single value
		2	Relevant characteristics
		1	Full sensor time series
24	us_prediction_sensor_num	Number of not used sensors (reference are available sensors of used dummy)	
25	us_prediction_sensor_relevance	Relevance of used sensors, grades, where 1 is best	
		4	Irrelevant
		3	Physics relevant
		2	Utilized in consumer tests
		1	Utilized in legislation
26	us_prediction_type	Value of prediction type, grades, where 1 is best	
		5	Binary classification (e. g. critical, uncritical)
		4	3 classes
		3	4-5 classes
		2	≥ 6 classes
		1	Regression
27	us_metamodel_num_sim_calibration	Calibration simulations per environment	
28	val_range	Width of validity range	

ANALYSIS METHOD FOR A TRAFFIC ACCIDENT USING MOTORCYCLE PROBE DATA

OSAMU ITO, TAKAYUKI KAWBUCHI, HIDEO KADOWAKI, YUJI TAKAGI1
Honda R&D Co., Ltd.
Japan

HIROKI TANAKA
Honda Motor Co., Ltd.
Japan

Paper Number 23-0136

ABSTRACT

To reduce the number of the fatalities among the motorcyclist in Asian countries, it is necessary to analyze and clarify the cause of the accident, however, the accident data are insufficient in these countries for the accurate analysis. To compensate for insufficient accident data, the authors approached to analyze the accident using the probe data obtained from vehicles.

The investigation was conducted by the riding data acquired from the 50 cc motorcycles, including the location information in 1 second cycle, the vehicle speed and the throttle opening signals in 0.2 seconds cycle acquired from the Global Navigation Satellite System (GNSS) and the Electronic Control Unit (ECU), respectively. The time historical data from GNSS and ECU were divided into 5798 trips, separated by the time interval longer than 1 minute. During all trips, there was only one accident. The acquired data were processed by the autoencoder model to extract the characteristics of the trips and riding behavior. The autoencoder model has the latent space between the encoder and decoder to analyze the trips and riding behavior. The information of trips and riding behavior in the latent space was quantified using Kernel Density Estimation to express the anomaly of the trips and riding behavior. In addition, riding simulations were conducted based on GNSS and ECU information to validate the results of abnormality detection by the autoencoder.

The results showed that the accident data were classified as abnormal behavior. The anomalies could be expressed as changes with time history. It proved that the riding abnormalities appeared 30 seconds before the accident occurred. When the simulation was also performed to reconstruct the accident, it was observed that the rider was riding dangerously such as slipping past the car or accelerating and decelerating rapidly.

The authors devised a method to analyze the causes of traffic accidents by using the autoencoder model and riding simulation. This method is expected to improve the efficiency of accident data collection and analysis in regions where accident data for motorcycles is lacking, such as in developing Asian countries.

INTORDUCTION

According to World Health Organization report [1], accidents involving motorcycles account for 28% of all traffic fatalities worldwide, which is second only to automobiles; countermeasures against traffic accidents involving motorcycles are an important research issue to reduce the number of fatalities. In general, to reduce traffic accidents, it is necessary to analyze the actual conditions of accidents and elucidate the causes of accidents. Since accidents involving motorcycles are more prominent in Asian developing countries such as Thailand [1], it is important to analyze accidents in these countries. Previous studies [2] showed the factors of motorcycle accidents by the investigation of the motorcyclist accident data in Thailand. However, the data contains the following residual issues and that precluded an elaborate analysis. First, the volume of the data was insufficient for the investigation and the accuracy of the analysis was low. Moreover, since various data aggregators, which are the police, hospitals, and insurance companies, investigated individually, it is hard to comprehend the relationships among respective data [3]. Second, the numerous accident investigation was engaged manually, which resulted in the inaccurate data collection due to the error in the descriptions or lack of information. Therefore, these manual works require the large amount of costs to ensure the data accuracy.

To collect more adequate data, it is necessary to create the efficient data collection process by eliminating the individual works. In recent years, the application of Intelligence Technology Systems has promoted the use of vehicle probe data and enable the effective estimation of road conditions and traffic accident risks [4]. In addition, Matsuo et al. improved the accuracy of collision risk estimation for vulnerable traffic by using probe data [5]. The objective of this study is the proof of the concepts applying the probe data obtained from motorcycles to traffic accidents analysis without any investigation reports by the third-party organizations.

METHODS

Motorcycle probe data

The investigation was conducted by the riding data acquired from the Global Navigation Satellite System (GNSS) instrument and the Electronic Control Unit (ECU) installed on the 50 cc motorcycles, including the location information in 1 second cycle, the vehicle speed and the throttle opening signals in 0.2 seconds cycle, respectively. The time historical data from GNSS and ECU were divided into 5798 trips, which was separated by the time interval longer than 1 minute. During the period of all trips, there was only one accident. A schematic diagram of the accident trip is shown in Figure 1.

Labeling rider behavior

Since riding behaviors cannot be directly observed from the probe data, those were estimated by the locations, the speeds, and the azimuth angles information. Estimated behaviors were defined as going straight, turning right, turning left, stopping, accelerating, decelerating, and cruising based on each state which were listed in Table 1, respectively. The turn direction was defined by the integration of the azimuth angle of travel per unit time within 5-second intervals. The state of acceleration or deceleration was defined based on the comparison of the speed of start and end within 5-second intervals with the average speed. For example, if the start speed was less than average speed and the end speed was greater than average speed, the behavior was defined as acceleration. All of the probe data was separated into 5-second intervals and labeled those riding behaviors according to the definition. In order to represent as various riding behaviors as possible, we defined 10 classes of riding behaviors: “straight + acceleration”, “straight + cruise”, “straight + deceleration”, “right turn + acceleration”, “right turn + cruise”, “right turn + deceleration”, “left turn + acceleration”, “left turn + cruise”, “left turn + deceleration”, “left turn + deceleration”, and “stop” by combining [straight, right turn, left turn] with [acceleration, deceleration, cruise] labels. These definitions allow classification of which riding behavior was being performed at a given time in the probe data. This enables analysis of riding behavior until the time of an accident.

Training model of riding history

Probe data contains a vast amount of data on normal riding. In order to analyze accidents, it is necessary to extract only information on the occurrence of accidents. Therefore, a classification model is constructed from the probe data, which can be regarded as the occurrence of an accident.

As the probe data contains a large volume of the data regarding a normal riding behavior without any accidents, it is necessary to extract the part in a short duration related to the traffic accident. The classification model is required to detect the rare incident from the data, however, since there is only one accident data in the probe dataset of this study, it is inappropriate to build the classification model by a supervised learning which generally requires many ground truth data. On the other hand, an anomaly detection model as an unsupervised learning is effective to detect the presence of the error incidents such as the traffic accident by a sparse ground truth data [6]. An anomaly detection model is possible to be trained by various types of data such as complicated images and time historical data [7]. Previous study built the autoencoder model to detect an anomaly taxi route by means of a large amount of vehicle trajectory data [8]. The autoencoder model contains a latent space connecting the input and output variables and the space is observable by visualizing the dimension-reduced vectors. The latent space is the mixture distribution consisting of the mean and variations, therefore, it is possible to determine whether the similarity of the newly obtained data is average or an outliers an anomaly by measuring the distance to a cluster of features in the latent space. This study built the deep anomaly detection model based on an autoencoder to extract the error incidents from a large amount of historical riding data, which were assumed as that contains the anomaly rider behavior occurring an accident.

Convolutional Neural Networks were used for the encoder and decoder [9]. To ensure that features can be well separated in the latent space, the decoder network was set up to split the trip and riding behavior labels. The trip portion was trained with Mean Square Error loss function, while the riding behavior label portion was trained with Cross Entropy loss function [9]. After training was completed, the riding behaviors were classified into 10-class clusters every 5 seconds. The target riding behavior can be judged as abnormal by measuring the distance from the center of the cluster (Figure 2). To quantitatively measure the distance in the latent space, Kernel Density Estimation (KDE) was used for each cluster [10]. Each cluster's center was defined from the mode of the KDE. The distance from the center was measured in Mahalanobis' distance [11]. For example, riding behaviors of a rider always near the center of the cluster can be considered normal riding, while riding behaviors far from the center of the cluster can be considered abnormal riding. By measuring this distance for riding behavior every 5 seconds, the degree of riding abnormality can be observed in the time history. Figure 3 and Table 2 show the schematic diagram and parameters of the autoencoder model, respectively.

RESULTS

Figure.4 shows the distribution of average Mahalanobis' distance during a trip. The average Mahalanobis' distance during a trip was most often between 0.4 and 0.6. On the other hand, the average Mahalanobis' distance for the accident trip was 1.43. Since the average Mahalanobis' distance was more than 1.4 within 5% of all trips, the anomalies can be classified. Figures 5 and 6 show the latent space and the time history graphs of Mahalanobis' distance for the accident trips. It was found that the Mahalanobis' distance increased about 30 seconds before the timing of the accident. In particular, the distance increased during the actions of accelerating straight, cruising straight, and decelerating straight.

DISCUSSION

In order to analyze how the rider was doing before the timing of the accident, the riding reconstruction simulation was conducted. Motorcyclemaker by IPG was used for the simulation [12]. The vehicle model was simulated only by the exterior shape, and location and time information was input to reconstruct simple riding. The roads were reconstructed by downloading Keyhole Markup Language files of the surrounding roads ridden from Google Maps and inputting them into Motorcyclemaker [13]. The objects such as sidewalks, buildings, traffic signals, and signs were reconstructed by using the 3D city model opened by the Ministry of Land, Infrastructure, Transport and Tourism [14] and applying textures to the objects with reference to Google Street View [15]. Figure 7 shows the picture of the riding trajectory on the road obtained by the simulation. Figure 8 shows the schematic diagram of the travel trajectory. These figures show that the vehicle seems to stop slightly behind the stop line at the intersection and then move forward on the roadway boundary before the intersection. After passing through the signal intersection, the vehicle was traveling at speeds fluctuating between 35 km/h and 40 km/h. Although the speed limiter limits the upper speed limit to about 40 km/h, the vehicle's riding behavior was unnatural, with repeated rapid acceleration and deceleration. Since the accident report noted the presence of a car ahead, we considered the rider to have repeatedly acted in a hurry to keep a short distance from the car ahead. Figure 9 shows the setup with the other vehicles, placed on the reconstruction simulation based on the above assumptions. From the results of the accident reconstruction simulation, the relationship between the Mahalanobis' distance time history and riding behavior is discussed and the results are shown in Figure 10. In addition, the capture of events between the time of arrival before the intersection and the occurrence of the accident is shown in Figure 11. In this accident case, the following three factors are the causes of the accident.

- The rider was slipping past the car at the intersection.
- The rider was accelerating and decelerating rapidly to keep a short distance from the vehicle ahead.
- The rider changed lanes and immediately returned to the original lane.

In this study, the anomaly detection model using the probe data was able to identify the abnormal riding behaviors that led to the accident. Furthermore, by conducting the simulation to reconstruct the accident, we were able to find the insights into the behavior just prior to the accident, which were not recorded in the accident reports. This allowed us to analyze riding behavior about accidents, without the need to conduct on-site investigations. However, the data in this study is limited and the number of accidents is small. We believe that expanding the collection of probe data and validating the methodology of this study will enable reliable analysis of traffic accidents involving motorcycle vehicles in the future.

REFELENCES

[1] World Health Organization (WHO). Global status report on road safety 2018: Summary (No. WHO/NMH/NVI/18.20); 2018

[2] Suriyawongpaisal P. KEY FACTS ON ROAD SAFETY SITUATIONS IN THAILAND 2012-2013; 2015

[3] Japan International Cooperation Agency (JAICA). Project Research "Road Safety Initiatives in Developing Countries", Final Report; 2016. Available at: https://openjicareport.jica.go.jp/pdf/12260915.pdf, Accessed August 2, 2022. (in Japanese)

[4] Najumudeen K, Bin MEA. Traffic accident analysis and prediction using the NPMRDS.; 2020.

[5] Matsuo K, Chigai N, Chattha MI, Sugiki N. Vulnerable road user safety evaluation using probe vehicle data with collision warning information, Accid Anal Prev.2022;165:06528.

[6] Aggarwal CC. An introduction to outlier analysis. In Outlier analysis, Springer. 2013;1-40

[7] Chalapathy R, Chawla S. DEEP LEARNING FOR ANOMALY DETECTION: A SURVEY: 2019. Available at: https://arxiv.org/abs/1901.03407, Accessed August 2, 2022.

[8] Liu Y, Zhao K, Cong G, Bao Z. Online Anomalous Trajectory Detection with Deep Generative Sequence Modeling; 2020 IEEE 36th International Conference on Data Engineering (ICDE),949-960.

[9] Goodfellow I,Bengio Y, Courville A. Deep Learning, MIT Press: 2016, Chapter 6 and 9.

[10] Chen Y. A Tutorial on Kernel Density Estimation and Recent Advances: 2017, Available at: https://arxiv.org/abs/1704.03924, Accessed August 2, 2022

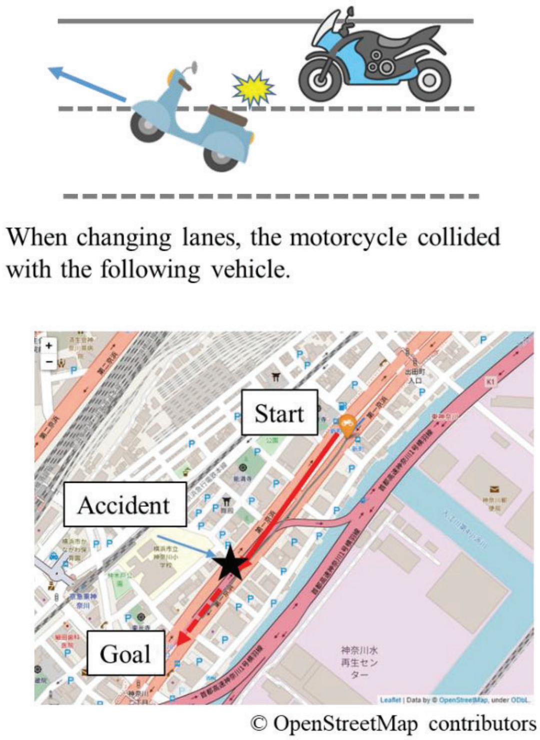
[11] Prasanta CM. On the generalized distance in statistics, Proceedings of the National Institute of Sciences of India.1936: 2 (1).

[12] IGP Automotive, Motorcyclemaker, Available at: https://ipg-automotive.com/, Accessed August 2, 2022.

[13] Google LLC. Keyhole Markup Language, Available at: https://developers.google.com/kml, Accessed August 2, 2022.

[14] Ministry of Land, Infrastructure, Transport and Tourism (MLIT), PLATEAU, Available at: https://www.mlit.go.jp/plateau/, Accessed August 2, 2022.

[15] Google LLC. Google Street View, Available at: https://www.google.co.jp/maps/preview, Accessed August 2, 2022.

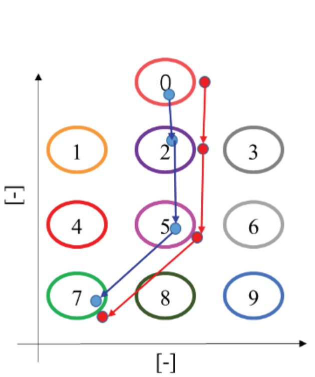

When changing lanes, the motorcycle collided with the following vehicle.

© OpenStreetMap contributors

Figure 1. Schematic diagram of the accident trip

Table 1. Riding behavior label defined from location information and velocity at 5 seconds intervals.

Label	Definition
Go Straight	The integrated value of the azimuth angle is within ± 20 deg.
Turn Left	The integrated value of the azimuth angle is under - 20 deg.
Turn Right	The integrated value of the azimuth angle is over + 20 deg.
Stop	The average velocity is under 5km/h.
Acceleration	Start speed is less than average speed and end speed is greater than average speed.
Deceleration	Start speed is greater than average speed and end speed is less than average speed.
Cruise	Other than acceleration and deceleration conditions.



Class	Behavior
0	Stop
1	Acceleration + Left
2	Acceleration + Straight
3	Acceleration + Right
4	Cruise + Left
5	Cruise + Straight
6	Cruise + Right
7	Deceleration + Left
8	Deceleration + Straight
9	Deceleration + Right

Figure 2. Schematic diagram of analysis method using latent space.

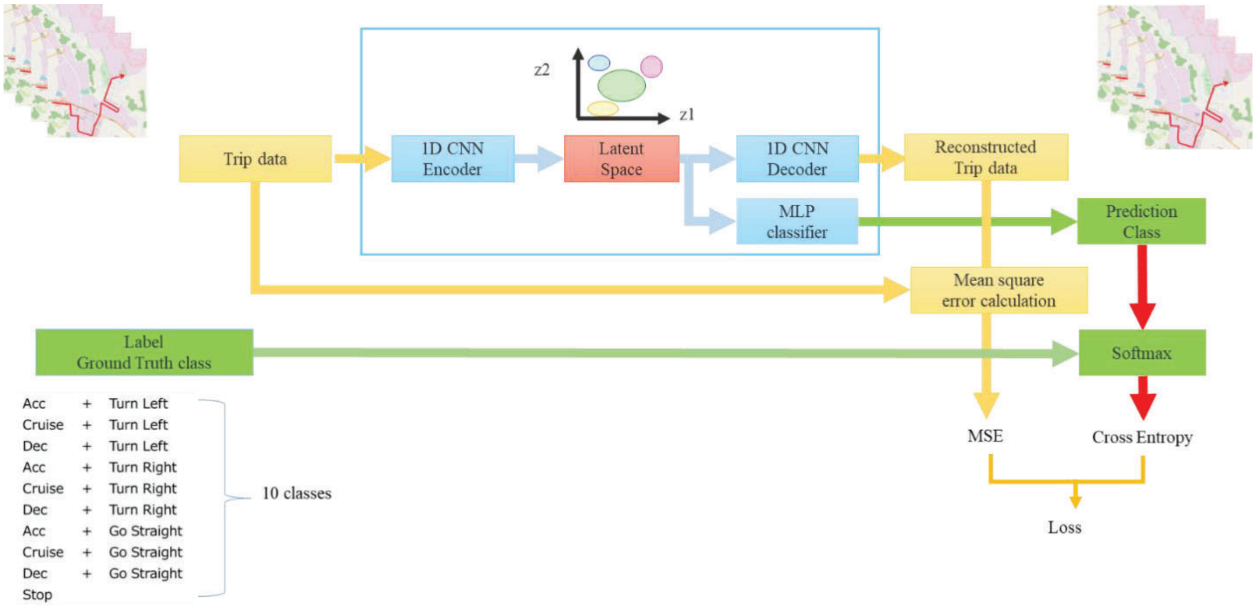


Figure 3. Schematic diagram of the Autoecndoer model

Table 2. Parameters of the Autoecndoer model

Encoder part

Layer	Cin	Lin	Cout	Lout	Kernel	Padding	Stride	Activation Function	Batch Norm
Conv1d	9	500	32	200	5	-	5	Relu	Use
Conv1d	32	200	64	32	5	-	5	Relu	Use
Conv1d	64	32	128	8	5	-	5	Relu	Use
Conv1d	128	8	256	1	4	-	5	Relu	Use
Linear	256	-	2	-	-	-	-	Tanh	Not Use

Decoder part

Layer	Cin	Lin	Cout	Lout	Kernel	Padding	Stride	Activation Function	Batch Norm
Linear	2	-	256	-	-	-	-	Leaky ReLU	Use
ConvTranspose1d	256	1	128	5	5	-	5	Leaky ReLU	Use
ConvTranspose1d	128	5	64	25	5	-	5	Leaky ReLU	Use
ConvTranspose1d	64	25	32	125	5	-	5	Leaky ReLU	Use
ConvTranspose1d	32	125	9	500	4	-	4	HardTanh	Not Use

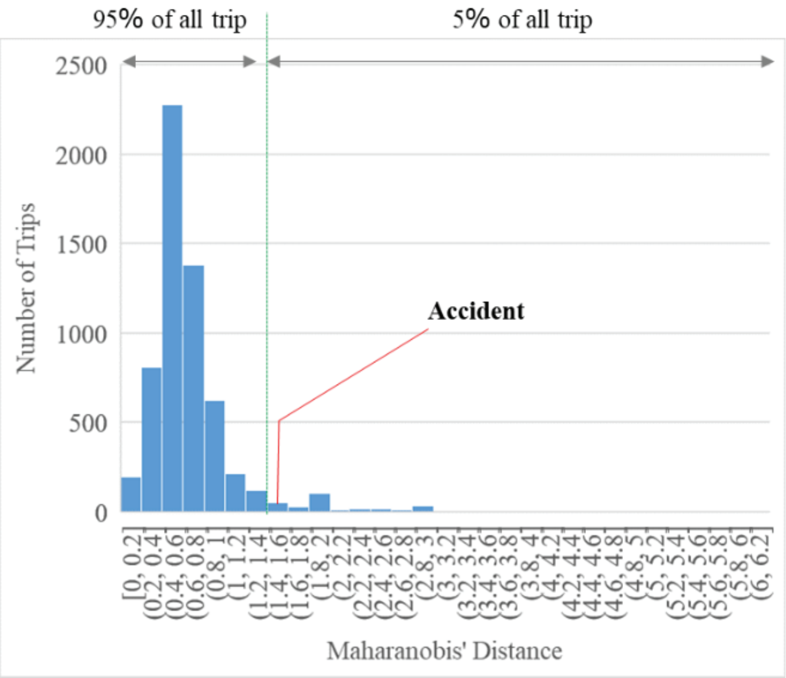


Figure 4. Histogram of average Mahalanobis' distance for all trips.

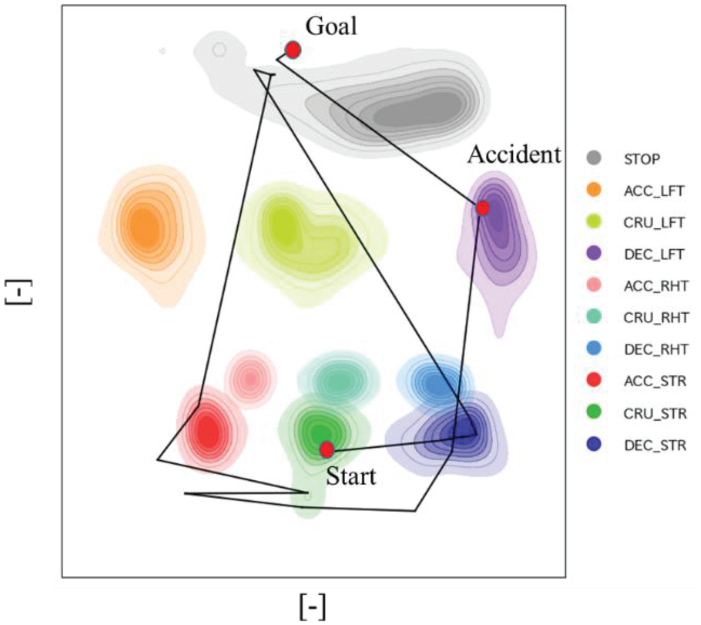


Figure 5. Trajectory of the accident trip for all trips in the latent space.

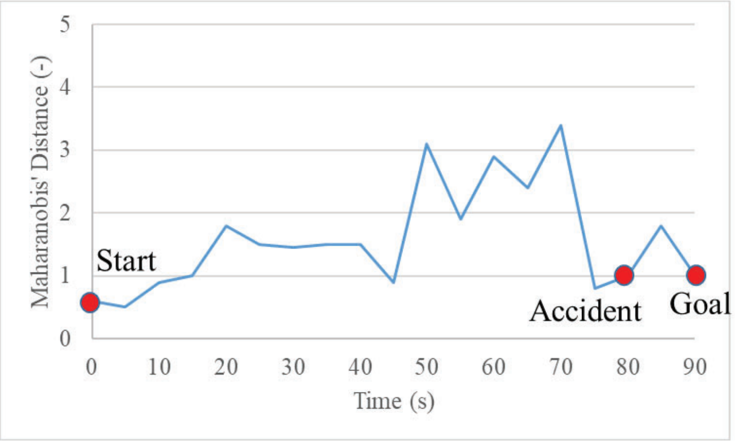


Figure 6. Maharanobis' distance of accident trip.

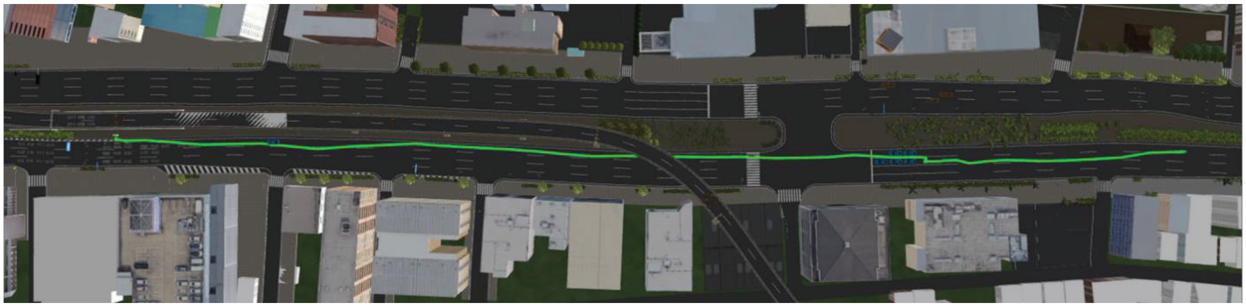


Figure 7. Riding trajectory on the road from reconstruction simulation

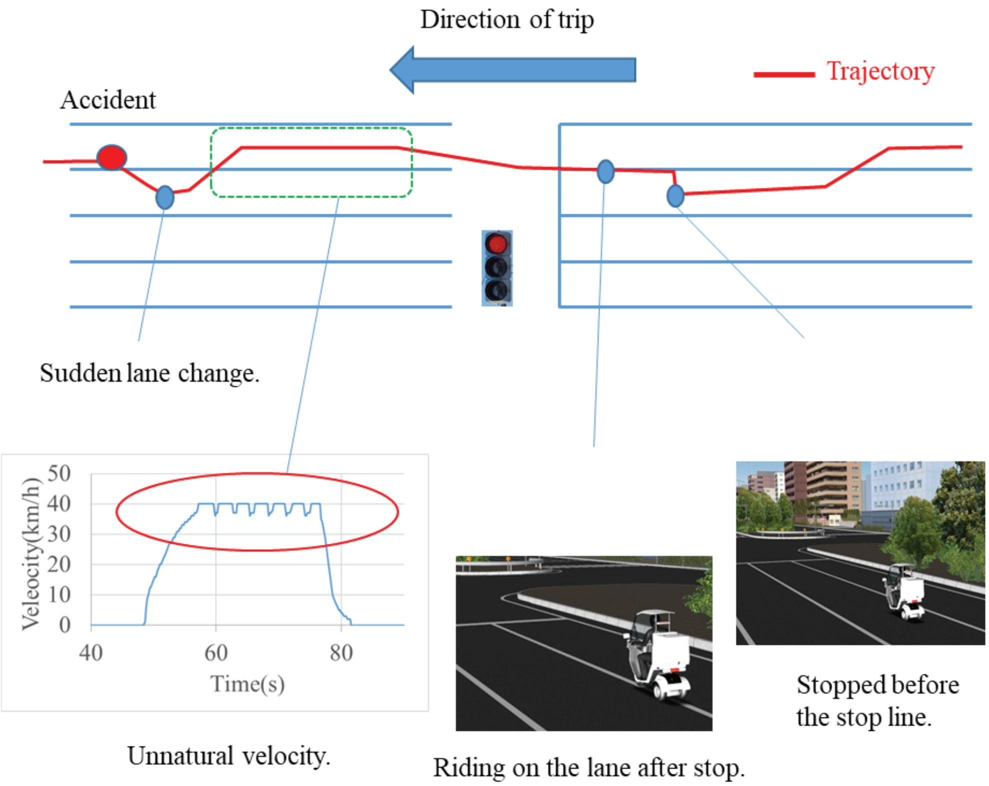
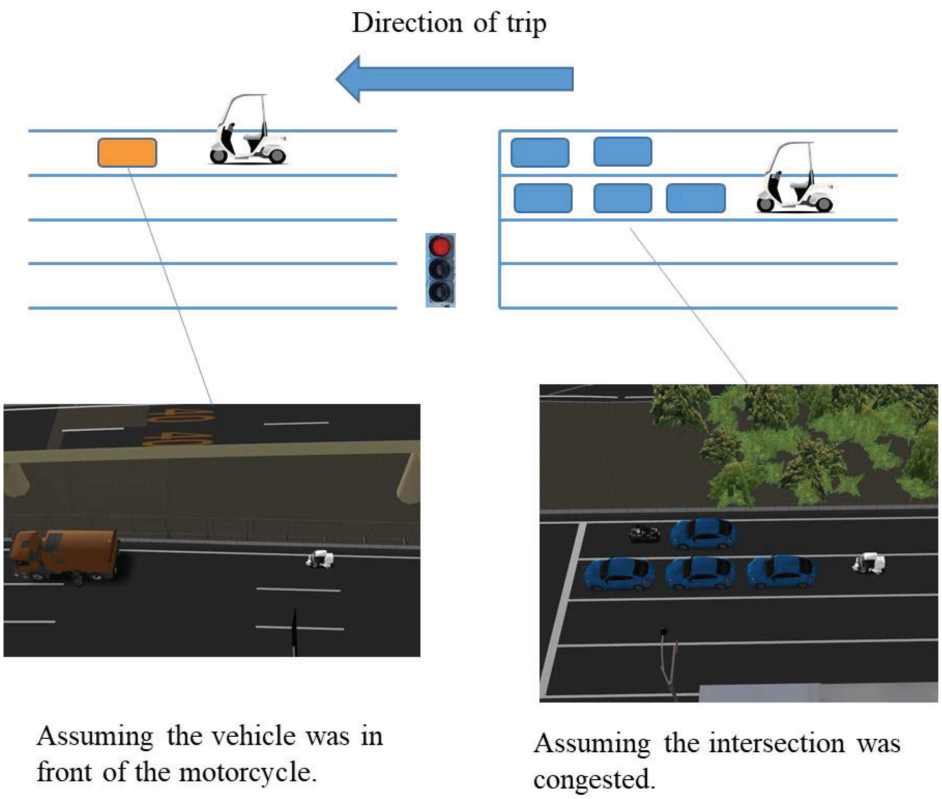


Figure 8. Schematic diagram of the riding trajectory with the features.



Assuming the vehicle was in front of the motorcycle.

Assuming the intersection was congested.

Figure 9. Assumptions for placement of other vehicles

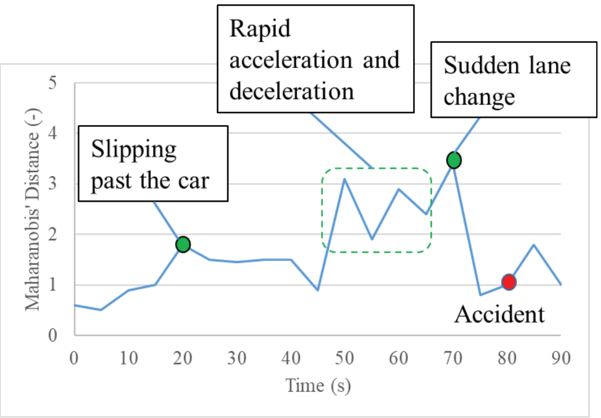
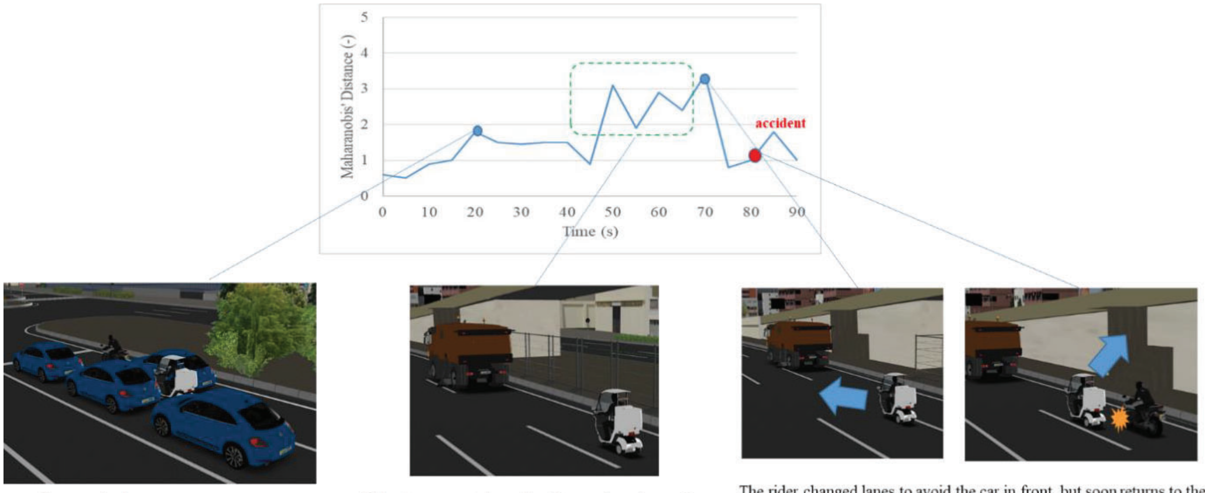


Figure 10. Relationship between riding events and Mahalanobis' distance.



Increased values
due to slipping past the car

Value increases to keep the distance from the car in front as close as possible.

The rider changed lanes to avoid the car in front, but soon returns to the original lane and collides with the vehicle behind.

Figure 11. Relationship between driving behavior and Mahalanobis' distance with simulation results

APPLYING AI METHODS ON VIDEO DOCUMENTED CAR-VRU FRONT CRASHES TO DETERMINE GENERALIZED VULNERABLE ROAD USER BEHAVIORS

Thomas, Lich
Jörg, Mönnich
Martin, Voss
BOSCH Accident Research, Corporate Sector Research and Advance Engineering, Robert Bosch GmbH,
70465 Stuttgart
Germany

Patrick, Lerge
Lennart V., Nölle
Syn, Schmitt
Institute for Modelling and Simulation of Biomechanical Systems (IMSB), University of Stuttgart,
70569 Stuttgart
Germany

Paper Number 23-0210

ABSTRACT

Urban traffic is characterized by limited traffic areas, varying traffic flows and the occurrence of different types of road users. To further advance automated mobility, the severity of injuries sustained by vulnerable road users (VRUs) in unavoidable accidents must be minimized. The project “ATTENTION”, supported by the German Federal Ministry for Economic Affairs and Climate Action, was set up to tackle this issue by developing a method for the real-time prediction of VRU injury risk using artificial intelligence (AI). The present study represents the first step in the ATTENTION project and evaluates behavioral aspects of VRUs in real-life car crash scenarios. Firstly, a comprehensive, hand labeled database of video documented VRU crashes from South Korean dashcams was set up. Secondly, the data was analyzed to determine relevant characteristics like pedestrian pre-crash movement and behavior. Afterwards a comparison against the German in-depth accident study database was performed. Finally, relevant scenarios were extracted, and AI-based preprocessing was applied. Body-shape-estimation methods were used to extract pedestrian poses and kinematics for further statistical processing.

In 9,724 video documented crashes, 369 frontal primary collision against VRUs were deemed usable. The analysis reveals that every 4th crash in this sample is potentially not avoidable due to physical limitations. The VRU recognized the car before impact in every 2nd crash, possibly performing evasive actions prior to first impact. Comparisons revealed that 31,000 similar car-VRU crashes were documented in the German In-Depth Accident Study (GIDAS) database. The estimation of plausible shapes and kinematics was possible in 37 of 319 pedestrian cases (12%), while 10 of 50 videos (20%) involving cyclist could be processed. Distinct pre-crash poses and kinematics were objectively identified and were shown to be different from standard gait-cycle kinematics. The VRU shapes and poses were used to define average pre-crash body shape appearances and hull-spaces for use in future human body model simulations.

The results of this study show that a VRU pre-crash behavior can be objectively determined from low-quality in-field video data using AI-driven methods and that it differs from regular human motion patterns. Furthermore, it shows that this video data can be used to setup a position and movement database. Both lay the foundation to estimate an injury risk index of VRUs in the later stages of the ATTENTION project.

INTRODUCTION

According to the World Health Organization (WHO), about 1.35 million people die in road traffic accidents each year, with pedestrians and cyclists representing 26% of global traffic related deaths. In other words, in one out of four fatal road traffic crashes, a pedestrian or cyclists is killed [1]. Thus, ensuring the safety of vulnerable road users (VRUs) became a focus for both original equipment manufacturers (OEMs) and suppliers. Safety solutions such as the automatic emergency braking (AEB) for VRU were developed in the past and are continuously assessed in consumer ratings such as the European New Car Assessment Program (Euro NCAP) [2]. But with increasing automation, even more complex traffic scenarios involving VRUs will need to be accounted for in the design of enhanced safety systems. Modern urban traffic is characterized by limited traffic areas, varying traffic flows and the simultaneous occurrence of different types of road users. To further advance automated mobility and to address potential safety concerns related to the lack of human supervision, the severity of injuries sustained by VRUs in unavoidable accidents must be minimized. The project "ATTENTION", supported by the German Federal Ministry for Economic Affairs and Climate Action, was set up to tackle this issue by developing a method for the real-time prediction of VRU injury risk using artificial intelligence (AI) methods. The present study represents the first step in the ATTENTION project and evaluates behavioral aspects of pedestrians and cyclists in real-life car crash scenarios in terms of objective measurements like their individual shape appearance and local joint angles in the Pre-Crash-Phase.

To this end, records retrieved from event data recorders (EDRs) as well as dashcam videos collected from Korean cab drivers were first selected based on a pre-defined set of criteria and used to supplement retrospectively collected accident data, as commonly employed methods to reconstruct collisions lack information with regards to the pre-crash behavior of each crash participant. Moreover, the dashcam videos provide valuable visual information about the pre-crash phase which grants a better understanding about the behavioral aspects of the driver and other participants in the accident. The present study is the first of its kind to analyze the pre-crash behavioral aspects of Korean pedestrians and cyclists involved in frontal car collisions. To assess the applicability of the Korean traffic data thus obtained for Germany, similar selection criteria were applied to the data gathered in the German In-Depth Accident Study project (GIDAS) which provides valuable detailed information about collision events and injury mechanisms [3].

Next, the dashcam videos were further processed through AI-based body-shape-estimation methods to determine pre-collision VRU positions comparable to Schachner et al. [4] which are categorized using objective criteria based on joint angle positions. The previously published approach to derive and evaluate 3-dimensional body shapes from high definition dashcam video data is both reliant on the selection of appropriate video input and the manual classification of resulting data. The purpose of this work is to show a more automated approach that is applicable to unfiltered video data irrespective of video quality and which generates physiologically more plausible results. In addition, it is shown, how the results of the presented method can be statistically postprocessed to get a more objective general view on VRU precrash shapes, hull-spaces and kinematics.

METHODS

Data sources

The video footage used for the precrash motion reconstruction in our current study is based on data provided by the Korean Transport Institute (KOTI). In cooperation with the Korean Mutual Taxi Association (KOTMA), more than 30,000 cabs were equipped with recorders capturing video and further vehicle data [5]. The operational area of the vehicles covered mostly urban areas in Incheon City near Seoul. The video stream consists of frames with 640x480 pixels (px) resolution at a frame rate of 5 frames per second (FPS). The vehicle position was captured using the cab's Global Position System (GPS) signal and was used to derive the speed of the vehicle. Longitudinal, lateral, and vertical acceleration within a range of ±10 m/s^2 at a 25 ms sample rate were continuously measured in addition to the video stream. In case of a collision, a minimum of 20 seconds of data-stream preceding the event were stored. Crashes of all severity levels are included, from accidents with property damage only, to crashes leading to personal injuries. No on-site investigation or any further reconstruction was done retrospectively. During an investigation period of 5 years, more than 30,000 crashes were recorded. The raw videos were provided within a collaboration of

Bosch Accident Research, KOTI and KOTMA in 2010/11. The data was provided anonymized with no personal information included. For the present study, 9,724 video documented cab crashes were available.

The study uses further in-depth accident data from the GIDAS database. Recordings contain detailed on-site information about the accident, location, and weather conditions, as well as involved parties considering more than 2,500 parameters per crash [3]. After the documentation, the cases were postprocessed and reconstructed. For the present study, we use a subset of the GIDAS database containing more than 40,000 crashes with personal injuries. To draw conclusions for Germany, GIDAS data is extrapolated using the type of crash, location, injury severity and vehicle age by applying the method described in Sulzberger et al. [6] and using data from the German Federal Statistical Office (DESTATIS) for 2021 [7][8].

Data selection

To synchronize the dashcam videos and the data from the GIDAS database, some criteria were defined and applied to both datasets. Following criteria needed to be fulfilled:

- Collision between a passenger car and pedestrian or cyclist (either 1st or 2nd participant)
- Primary impact against pedestrian or cyclist
- Impact point at the front of the vehicle (CDC2=1)[1]
- Car only moves forward (≥5 km/h, excluding standstill or backwards driving)

This allowed for the extraction of crash relevant parameters from the GIDAS database which is required to setup the parameter space for future crash simulations. After applying the listed selection criteria to the 9,724 video documented crashes, 369 frontal primary collision against VRUs were deemed usable. The selected relevant dashcam videos were processed further.

Data labeling and annotation

Firstly, a comprehensive, hand labeled database of video documented VRU crashes from South Korean dashcams was set up. Initially, the video data was not labeled or annotated, nor was further on-site investigated information about the crash available. Therefore, manual annotation of the crash sequence and extraction of relevant crash parameters was required. The capture process of various timestamps throughout the collision event is shown in Figure 1. Details about the database are available in Lich et al. [9].

Data analysis of the pre-crash behavior of VRUs

In the next step, we further analyzed the pre-crash behavior of VRUs in the remaining 369 videos based on the hand-labeled data. As the reaction of the VRUs is captured in the video image, we were able to perform a rough subjective classification. It was marked whether the VRUs perceived the vehicle before the collision or not, while the type of reaction was classified afterwards. Further details derived from the videos as well as descriptive statistics can be found in Lich et al. [9]. The VRU poses extracted from the dashcam videos will serve as time dependent target positions for the crash simulation which will follow in the later stages of the ATTENTION project, since they represent natural in crash behavior.

To describe the parameter space for the crash simulation in more detail, we analyzed the GIDAS database. To describe the pre-collision phase and in addition to commonly considered parameters such as initial- and collision speed, deceleration and impact location, the collision angle between pedestrian or cyclist and car was determined.

[1] Collision Deformation Classification (CDC) according to the National Highway Traffic Safety Administration (NHTSA)

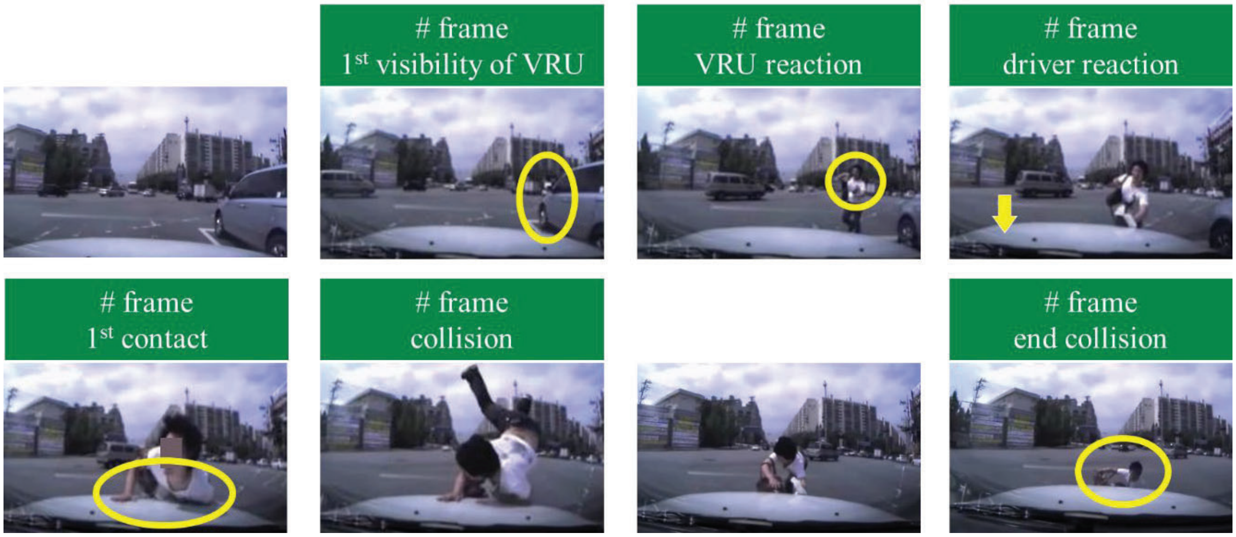


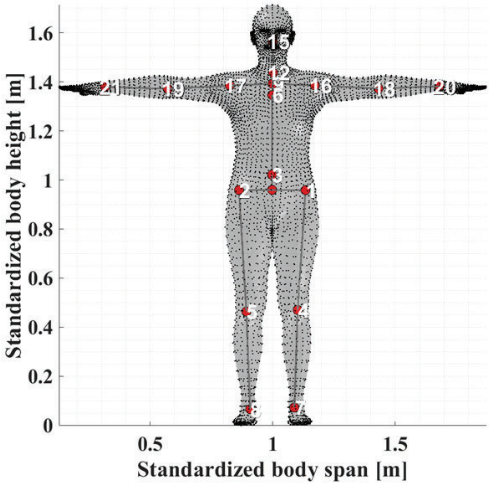
Figure 1: Capture process of various timestamps throughout the whole collision event

AI-driven Preprocessing

In parallel to the subjective analysis of the video data, an AI-driven video preprocessing was performed. First, the preselected video data was digitally enhanced to a four-times higher resolution of 2560x1920 px, with sharper contrast and less noise by applying the openly available AI-based enhancement algorithm "Real-ESRGAN" [10] with the "realesrnet-x4plus" model on all 369 videos.

These enhanced videos were processed further by applying the pose and shape estimation wrapper "PARE" [11]. PARE was chosen because it achieves a more plausible and robust estimation result when faced with partial occlusions caused by the car front or environmental structures. In this wrapper, first the 2D pose estimation software "OpenPose" [12] was applied to the videos with a tracking net resolution of a quarter the size of the video resolution. Afterwards, the resulting 2D joint locations are used as the initial condition to perform the 3D shape estimation with SMPL [13] by using the default optimization weights.

The internal skeleton regressor "SPIN", which was introduced by Kolotourus et al. [14], was used to make the underlaying skeletal structure as biofidelic and comparable to conventional motion tracking results like the H36M dataset [15] as possible. However, a modification was made to the orientation of the PARE output. The initial orientation of the internal functional skeleton was defined based on the neutral zero position instead of the original T-pose. This resulted in related movements in the joints, such as elbow and knee flexion, now being performed about the same axis. In addition, the clavicle joints and the shoulder joints have been combined into one and are only resolved in the shoulder joint. Also, the coordinate directions of the local joint coordinate systems were changed. The X-axis was made parallel to the ventral axis, the Y-axis parallel to the left-lateral axis and the Z-axis parallel to the cranial axis. All joints and their specific IDs are shown in Figure 2.



Joint ID	Joint Title	Joint ID	Joint Title
1	Left Hip Joint	12	Neck Joint
2	Right Hip Joint	15	Head Joint
3	Spine 1 Joint	16	Left Shoulder Joint
4	Left Knee Joint	17	Right Shoulder Joint
5	Right Knee Joint	18	Left Elbow Joint
6	Spine 2 Joint	19	Right Elbow Joint
7	Left Ankle Joint	20	Left Wrist Joint
8	Right Ankle Joint	21	Right Wrist Joint
9	Spine 3 Joint		

Figure 2: SMPL body shape model with SPIN joints for kinematic output in T-pose

For the results, only the shapes and poses of the crash partners were exported by manually eliminating other entities and false positive estimations that where still present after the enhancement. Apart from this, no manual correction or modification was used. The complete preprocessing workflow from the original frame to the kinematic export is exemplarily shown in Figure 3.

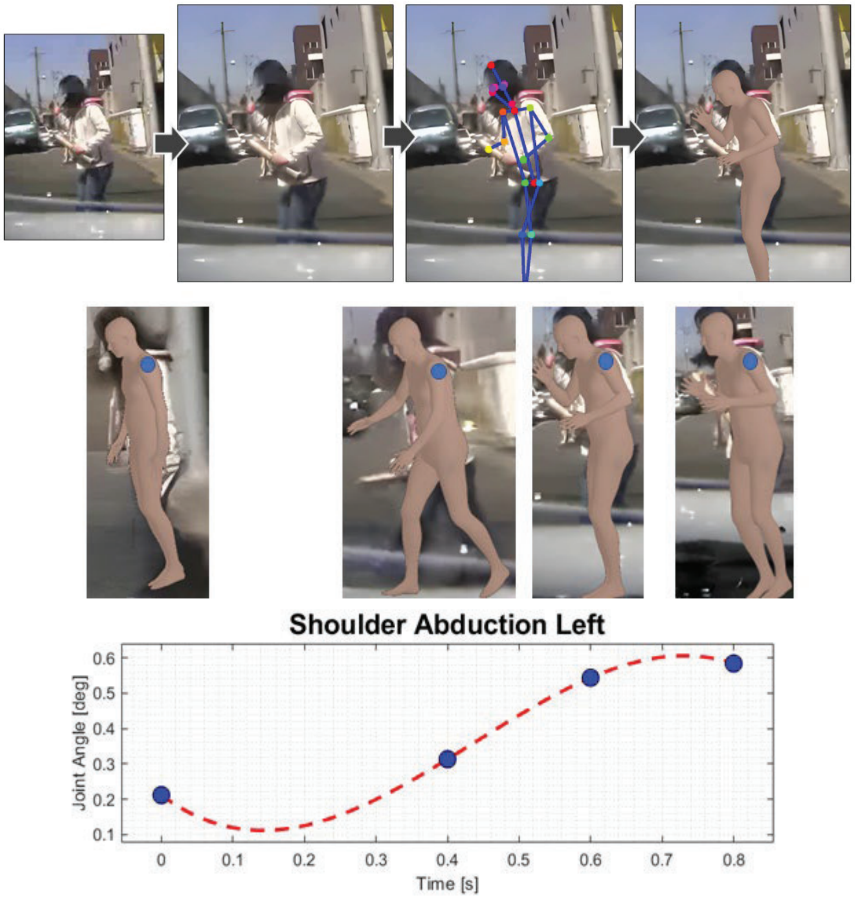


Figure 3: Preprocessing workflow: Enhancement, 2D estimation, 3D shape estimation and kinematic export

Postprocessing

In the last step, the results where statistically processed. This includes the determination of a normalized motion space over all detected entities of one type of VRU, the derivation of groups with recurring poses right before the crash, the determination of a mean shape for each group and the calculation of the average joint angles for the defined groups.

To be able to relate the appearances determined within the pose and shape estimation process to each other, a normalization of the external shapes M_N was performed. To make the outer shapes in non-local perspective comparable, all global hip-orientations (root body of the internal kinematic tree) where normalized to a 0°-orientation and all 6,890 nodes of the discretized surface M were modified with the same transformation by multiplying their XYZ-Coordinates with the 3x3 inverse matrix of the world-to-pelvis joint rotation matrix R_0 (Equation (1)), which is part of the pose estimation result.

$$M_{norm_N} = inv\left(R_{0,N}\right) * M_N \qquad \text{Equation (1)}$$

For the determination of the absolute motion space, the discretized surfaces were reduced to the normalized nodal points, and these were projected together into a three-dimensional space. Subsequently, a three-dimensional convex hull with about 400 vertices was placed around the point cloud formed in this way. For the determination of the convex hull, the method according to Barber et al. [16] was used.

Thus, a theoretical motion space could be identified, which, applied backwards to each individual orientation of an entity in relation to the world, provides information about a possible collision volume around a person for which only the orientation and movement of one point (in this case of the hip) must be predicted.

In the next step, the behavioral groups were determined. For this, the local joint angle differences from the final frame right before the impact were calculated first. To do so, each of the i = 23 joints of one entity ‘m’ was compared to its equivalent of another entity ‘n’ by determining the absolute angular difference for each of the 3 degrees of freedom (DOF) joints ($\theta_{diff_{i,mn}}$) using the 3x3 rotation matrices R_i that represent these local joint angles as shown in Equation (2).

$$\theta_{diff_{i,mn}} = \cos^{-1}\left(\frac{tr\left(R_{i,m}R_{i,n}^{T}\right)-1}{2}\right) \qquad \text{Equation (2)}$$

This lead to a difference-vector with i = 23 elements for every entity-to-entity relation, which was reduced to a mean angular error using the Root-Mean-Square method in the next step. This, for N entities, resulted in an error-matrix of the dimension NxN and was finally normalized by dividing the mean error values by the maximum possible error of 180°, which created the normalized error-matrix θ_{error} (Equation (3)).

$$\theta_{error_{mn}} = \frac{1}{180°}\sqrt{\frac{1}{k}\sum_{i=1}^{k=23}\theta_{diff_{i,mn}}^{2}} \qquad \text{Equation (3)}$$

Finally, network analysis was used to create a network comprised of those entities, whose error was non-zero and less than a variable generalized threshold τ. To find appropriate values for τ , it is recommended to perform an optimization to identify the value at which the number of groups of entities with more than one member is at a maximum and the number of entities without connection is at a minimum.

In the next step and based on the groups found this way, an average shape regarding every entity in one group was determined by calculating the average 3D-Position of every surface node over all entities in this group ‘G’. The number of existing connections from the network ‘deg’ was also included as a weighting factor (Equation (4)).

$$M_G = \frac{1}{\sum_{i_G}^{N_G} deg_{i_G}}\sum_{i_G}^{N_G}\left(deg_{i_G}\, M_{norm_{i_G}}\right) \qquad \text{Equation (4)}$$

In this way, the number of G discretized shapes could be determined for the different final appearances in the pre-crash phase. However, with the data available in this study, not only the generalized outer shapes but also group-specific average joint angles could be determined by using the output of the functional skeleton, which is a function of the body shape. For this purpose, the i = 23 joint rotation matrices of each entity in a group N_G were converted into quaternion form. This led to N_Gx23x4 sized matrices. Then, the number of occurrences of each entity in the matrix was multiplied according to its number of connections in the group-network/ level of 'deg' to get the same weighting effect as it was taken into account for the average shape.

These extended quaternion matrices were further averaged according to Markley et al. [17] which finally resulted in a matrix of size 23x4 for each individual group. At this point, the approach via quaternions had to be taken, since the supposedly intuitive cardan angles are not explicit due to sequencing in XYZ rotations, but are instead corrupted by previous rotations in their sequence and can therefore not be used for elementwise calculations. This matrix was then transformed back into rotational matrix formulation of size 23x3x3, as well as into a sequential X-Y'-Z'' cardan rotation.

The post-processing methods described above were applied to both the reconstructed videos of pedestrians and cyclists. However, both groups were always considered as separate types of VRUs and as such processed independently.

RESULTS

Field of effect

Applying the previously outlined selection criteria to the GIDAS database and extrapolating it towards the entirety of German traffic reveals that the selected Korean traffic scenarios correspond to about 29,000 similar car-VRU crashes with personal injuries in Germany. Overall, this is a share of about 11% of all car-VRU front crashes with personal injuries in Germany in 2021. Table 1 shows the selection process for both data sources.

Table 1: Available data and relevance for Germany

Criteria	Number of video-documented crashes (Korea 2010/11)	Estimated represented number of crashes in Germany (2021)
All crashes w/ personal injuries	9,724	258,987
... and car involvement	9,563	192,221
... and pedestrian or cyclist involvement	641	~ 66,000
... and first collision w/ cyclist or pedestrian	641	~ 60,000
... and car, pedestrian, cyclist as 1./2. participant		~ 59,000
… and frontal car collision (CDC2=1)[2]	375	~ 30,000
… and car collision speed min. 5 km/h incl. unknown (Field of effect of pfp ATTENTION)	369	~ 29,000
… out of pedestrian crashes	319	~ 8,000
… out of bicycle crashes	50	~ 21,000

Result Pre-crash behavior for pedestrians from video-data

For pedestrians we observed that in about 43% of the 319 sample the vehicle was not recognized prior to the collision whereas in 48% the pedestrian saw the car as shown in Figure 4. Once the pedestrian perceived the car, some reaction takes places. The decision-making process was not further evaluated. Moreover, the aim was to determine the parameter space which is needed to cover pedestrian crashes on a wider scope. However, in 38% of these cases, the pedestrian recognizes the car but did not change his behavior. Another 31% stopped or slowed down, while 16% sped up. In 10%, other movements took place, and in 3%, a defense posture was observed. These findings were further confirmed once the poses were extracted out of the video images.

[2] Collision Deformation Classification (CDC) according to the National Highway Traffic Safety Administration (NHTSA)

In Lich et al. [9], the time to collision (TTC) was determined. As a result, it was found that in about 25% of the accidents the time to collision is less than 1.2 seconds. This means that in one out of four car-pedestrian accidents, the collision cannot be prevented in this observed sample.

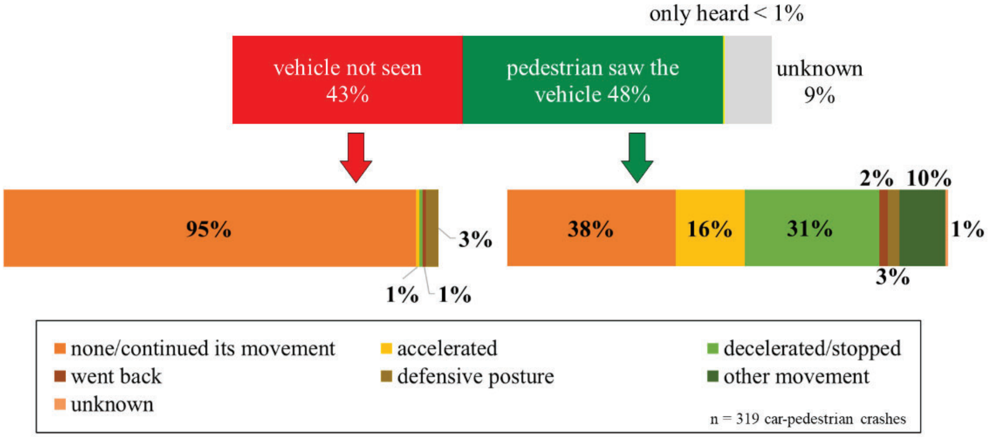


Figure 4: Pedestrian reaction by recognition prior to the impact determined out of 319 video documented car-pedestrian crashes

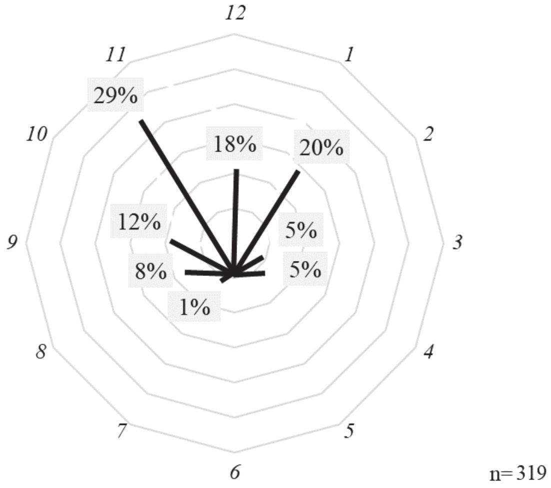


Figure 5 : Pedestrian impact direction determined out of 319 video documented car-pedestrian crashes

The impact directions of the pedestrians against the cars are shown in Figure 5. Most of the front crashes occur at impact positions of 11 o'clock (29%), 1 o'clock (20%) and full frontal at 12 o'clock (18%).

Result Pre-crash behavior for cyclists from video-data

In the sample of 50 car-cyclist crashes, 42% of cyclists did not see the car coming as in some of the crashes, the cyclists were hit from behind and thus were not able to recognize the car at all. However, 52% of them perceived the car but showed no reaction (77%). Some of them accelerated (8%) to avoid the collision and another 8% evaded mainly by steering intervention. The remaining 8% performed a mixed motion of steering and evading. Figure 6 shows the findings for the cyclist's pre-crash behavior.

Furthermore, for car-bicycle accidents, it was found that in about 18%, the TTC is less than 1.2 seconds. This means that in one out of five car-bicycle accidents, the collision cannot be prevented in this observed sample. The TTC was evaluated in detail in Lich et al. [9].

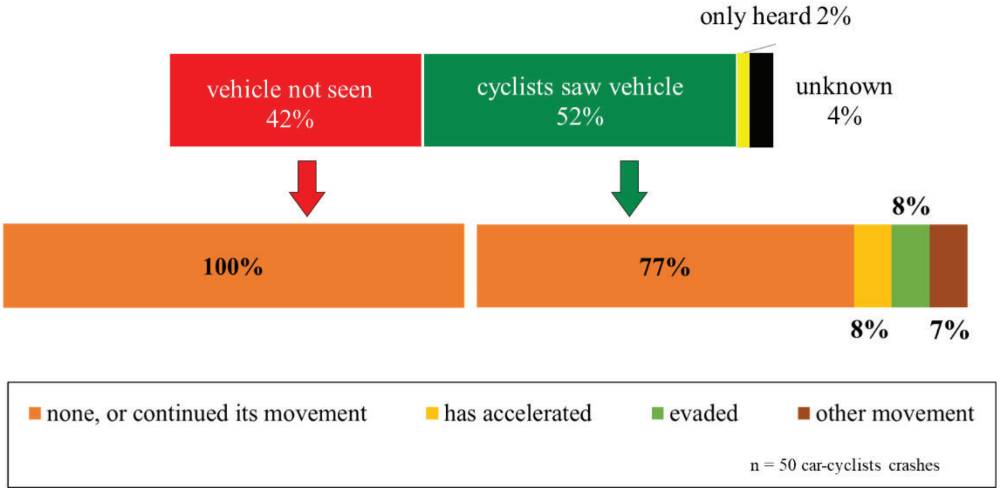


Figure 6: Cyclist reaction by recognition prior to the impact determined out of 50 video documented car-cyclist crashes

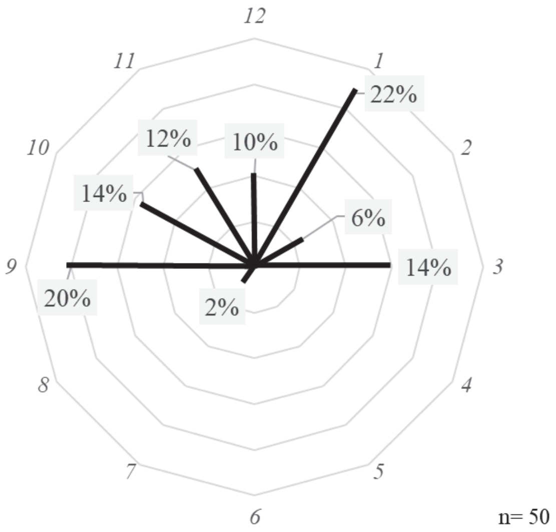


Figure 7: Cyclist impact direction determined out of 50 video documented car-cyclist crashes

The impact directions of the cyclists against the cars are shown in Figure 7. The majority of crashes occur at 1 o'clock (22%). It can also be seen that in 1 out of 3 crashes, the cyclists crossed the path of the vehicle laterally.

Results Enhancement

The initial use of an enhancement algorithm led to an improvement in pose and shape estimation such that the initial two-dimensional detection of people using OpenPose started on average 2 frames earlier (0.4 s; ~50% of total length) as shown in Figure 8. At the same time, the number of false positive detections decreased by ~20% due to the AI-assisted resharpening so that longer and qualitatively more reliable kinematics were achieved.

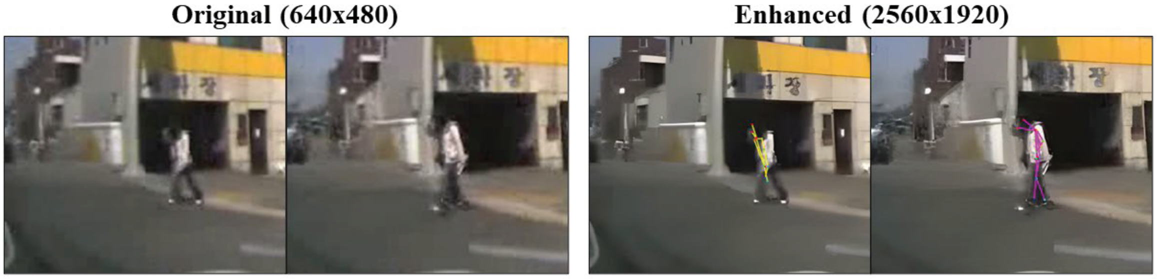


Figure 8: Effect of video enhancement on 2D pretracking with OpenPose

Results Person specific precrash shapes and poses

Starting from the database of preselected videos that represent our selected load case, an acceptable estimation of plausible shapes and kinematics was possible in 37 of 319 pedestrian cases (12%), while 10 of 50 videos (20%) involving cyclist could be processed. As illustrated in Figure 9, the behavior of natural individuals right before a crash with a vehicle, as they were estimated with the presented toolchain, is highly diverse. Most of the randomly selected individuals here show a behavior that is different from natural movements beside accidents like a natural locomotion. It is therefore highly advisable to define objective conditions that can be used to define and classify the different types of behavior in a more simplified yet appropriate form.

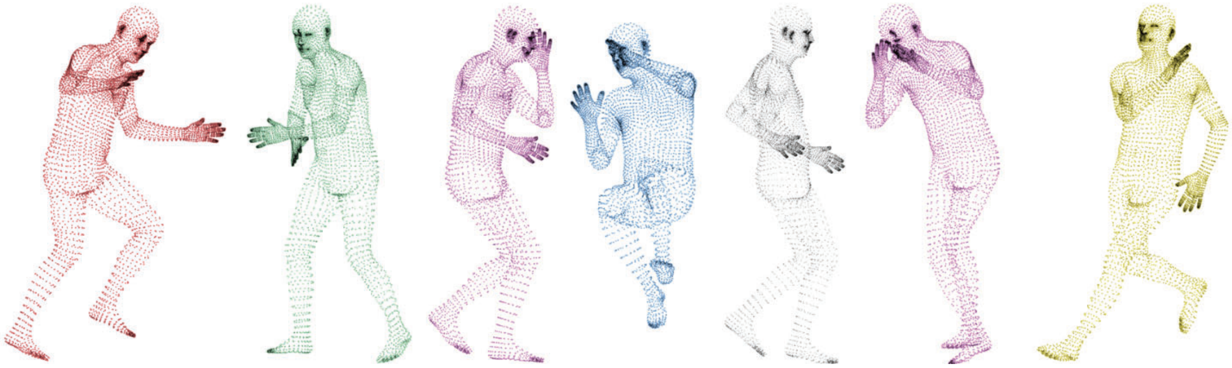


Figure 9: Examples of non-classified pedestrian postures prior to the collision derived from the video documented crashes

Results Hull-Space Pedestrian

Since the individual appearances, as shown in Figure 9, vary widely and appear to be unequally directed, the shape orientation was normalized in the first step of statistical processing. By applying the same origin point to all entities, a situation dependent maximum extension of the human body can be determined, which is unequal to the general maximum possible expansion of the human body, without the need to consider local joint angles in the first place. The resulting motion/hull-space as a result of all our reconstructed shapes is displayed in Figure 10. The hull allows for a rough estimation up until which point a collision is possible and where it is unlikely, only considering the hip centre position, size scaling and hip orientation.

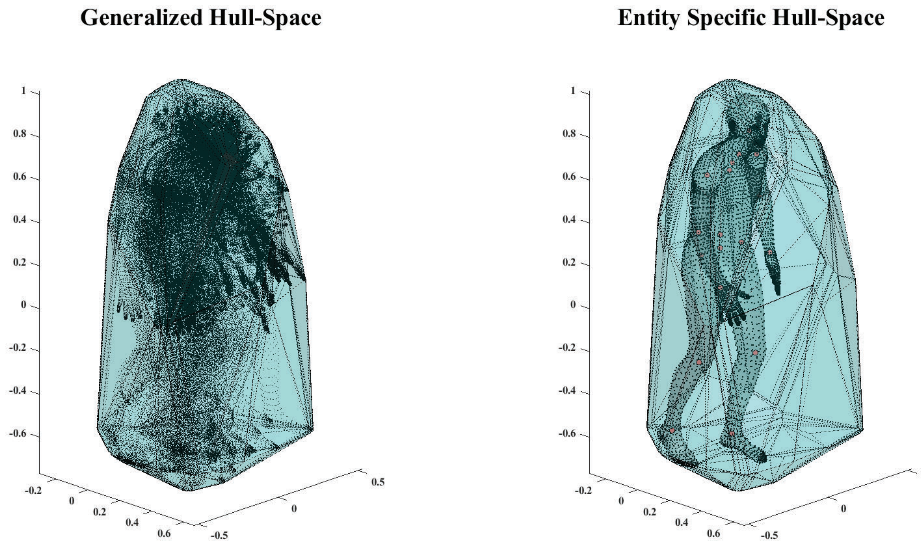


Figure 10: Observed hull-space over all pedestrians and for one representative example shape

Results Grouping Pedestrian

Due to the still very different moving appearances, a method was applied to group the reconstructed shapes and the associated movements and to generalize the external appearance as well as the movement at the level of joint angles in a group-specific way. These mean shapes and poses should still be diversified enough to be used for the further definition of pre-crash target positions for simulations with human body models.

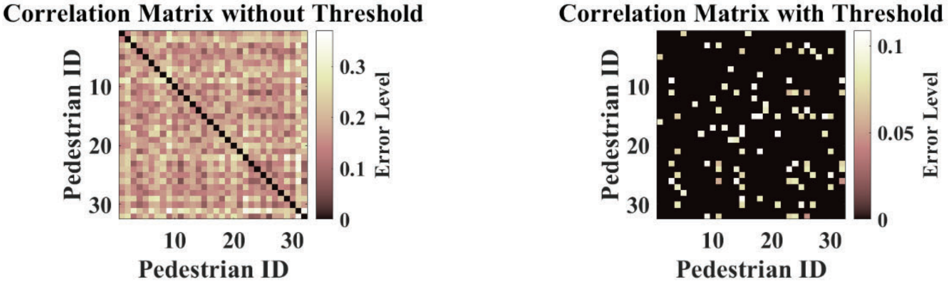


Figure 11: Pedestrian correlation matrix

The matrices shown in Figure 11 are quasi-correlation matrices there a larger cell value represents a larger mean difference in the joint angles and thus a stronger deviation in the outer appearance. The general purpose of these matrices is to find entity relationships with a high level of similarity, which was implemented by applying a threshold that annulled all cells with a larger error relation level. In the present case, a uniform threshold of 10.9% was used, as this fulfilled the criterion for a maximum number of groups with more than one entity at a maximum group size. The mentioned groups are shown in the following Figure 12.

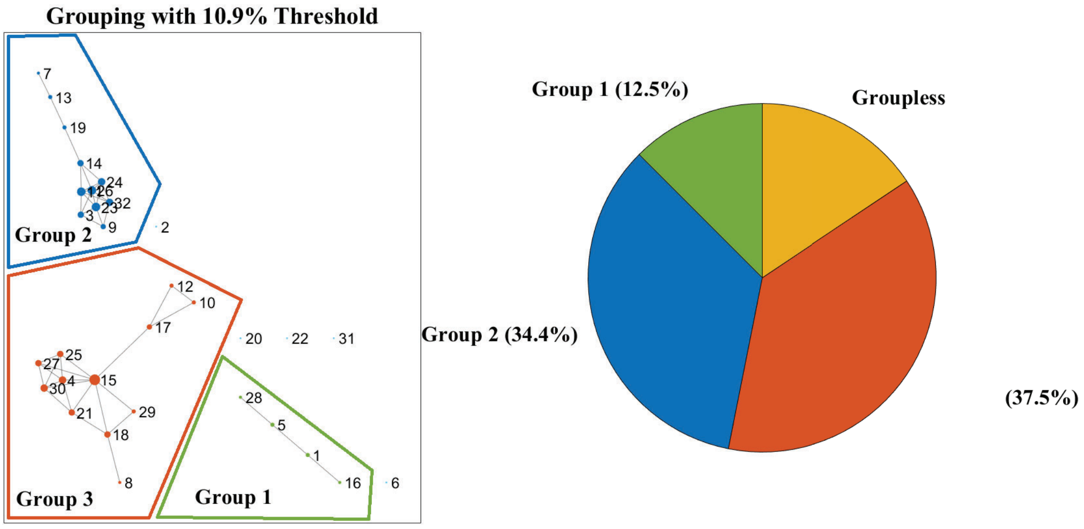


Figure 12: Resulting groups of pedestrian behavior

Figure 12 visualizes the distribution of the pedestrian entities in relation to each other as a graph and as a pie chart. With the selected threshold, three larger groups can be identified that are self-contained and thus visually distinguish themselves from the other appearances. Within these groups, there is at least one entity that has a particularly high similarity to most of the other group members, which should be taken into account in the further averaging of the body shapes.

It can be observed that the formed groups are not uniformly sized. Accordingly, there is a more frequent and a less frequent behavior in a crash situation. For example, group 3 comprises 37.5% of all pedestrian cases, while group 1 comprises only 12.5%. In total, 84.4% of the reconstructed cases were captured via the grouping method, which were subsequently included in the following shape and pose normalization.

Results Shape and Pose Pedestrian

In the following Figure 13, only the mean outer shapes and the joint locations are shown. The influence of each group entity was weighted depending on the number of its connections in the network.

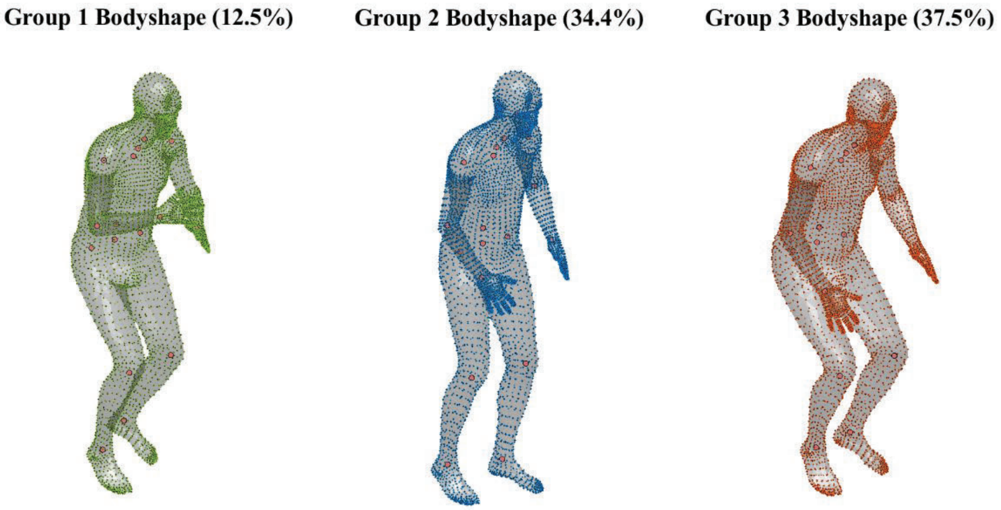

Group 1 Bodyshape (12.5%)
Group 2 Bodyshape (34.4%)
Group 3 Bodyshape (37.5%)

Figure 13: Mean pedestrian pre-crash shapes for groups larger one subject

Compared to Figure 9, the illustrations in Figure 13 show a much more symmetrical and less distorted appearance. The smallest group, consisting of 4 people (12.5%), is characterized by the fact that the respective entities have adopted a protective or defensive posture with their arms up and an angled leg position.

The average shape of the members of the largest Group 3 however, shows a body posture that corresponds more to a fright, preparation, or flight posture. It is characterized by angled legs like in group 1 but with a smaller inside angle as well as downward pointing and opened arms.

In contrast, the posture of group 2 barely deviates from a natural body posture. It can therefore be assumed that in at least 34.4% of our cases, there was no or barely any preparatory reaction to the impending accident.

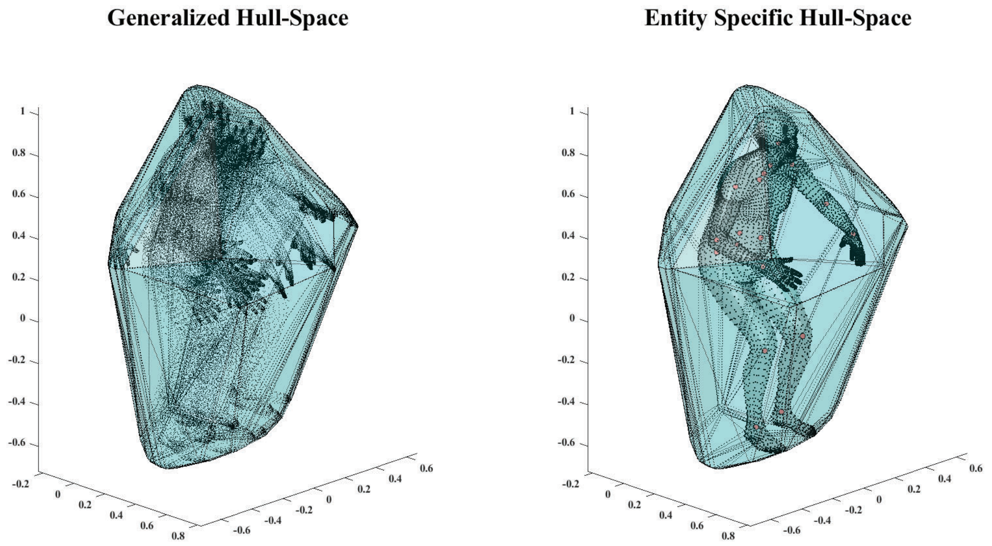
Table 2: Pedestrian mean pose joint angles

	Group 1			Group 2			Group 3		
JOINT ID	**X [°]**	**Y' [°]**	**Z'' [°]**	**X [°]**	**Y' [°]**	**Z'' [°]**	**X [°]**	**Y' [°]**	**Z'' [°]**
1 (L Hip)	5.84	-24.14	0.68	1.40	-17.79	0.21	8.52	-37.77	2.56
2 (R Hip)	-4.16	-30.94	-1.94	-3.58	-10.81	-1.09	-6.94	-39.52	-5.01
3 (Spine 1)	0.24	20.03	-0.96	0.70	14.75	0.91	0.56	25.31	-0.31
4 (L Knee)	-7.87	54.19	2.03	-4.27	24.39	0.50	-14.05	61.94	9.74
5 (R Knee)	9.18	51.38	-5.45	2.22	24.90	1.45	20.97	62.55	-18.83
6 (Spine 2)	0.22	1.41	-0.01	0.27	2.08	0.33	0.34	1.19	0.46
7 (L Ankle)	-6.44	-1.31	10.52	-6.22	0.49	8.43	-5.26	-0.58	10.14
8 (R Ankle)	6.42	1.02	-8.33	6.53	1.79	-7.79	5.39	4.01	-8.21
9 (Spine 3)	-0.69	2.72	-0.76	-0.37	3.12	0.59	-0.64	3.23	0.02
12 (Neck)	0.71	-3.21	0.14	-0.24	-0.11	-1.41	-0.30	-1.69	-3.97
15 (Head)	-0.52	4.71	2.61	-1.23	3.23	0.57	0.96	2.15	-3.35
16 (L Shoulder)	17.65	-34.67	-39.10	19.93	-11.43	-22.13	21.29	-26.18	-28.92
17 (R Shoulder)	-19.06	-35.67	35.67	-16.40	-16.90	26.93	-18.22	-25.08	30.78
18 (L Elbow)	43.10	-75.92	24.56	0.19	-49.20	-12.44	21.29	-67.69	4.93
19 (R Elbow)	55.88	-110.06	75.13	3.01	-47.71	14.55	-10.22	47.39	2.03
20 (L Wrist)	10.30	-3.99	2.71	8.44	-4.67	2.27	11.44	-3.87	0.02
21 (R Wrist)	-11.63	-5.19	-3.37	-9.55	-4.29	-2.03	-10.69	-3.21	1.01

The scaling invariant joint angle description of the three poses is listed in Table 2. Again, a weighting was applied to the determined pose data. These calculated joint angles are listed in the cardan notation for easier interpretation.

The values listed in Table 2 allow for an additional check for plausibility by comparing the angles at the supposedly natural joints in the hip, knee, ankle, shoulder, and elbow with limit values from the literature such as those in Barter et al. [18]. In this case, all joint angles passed this test. The reconstructed shapes and poses determined from them are therefore physiologically plausible.

Results Hull-Space Cyclist
After the highly diverse pedestrians, the hull and grouping methodology was also applied to the reconstructed data of cyclists. However, it neglects the frame of the bicycle which is not estimated during the pose and shape estimation but still a contactable object and effectively connected to the rider.

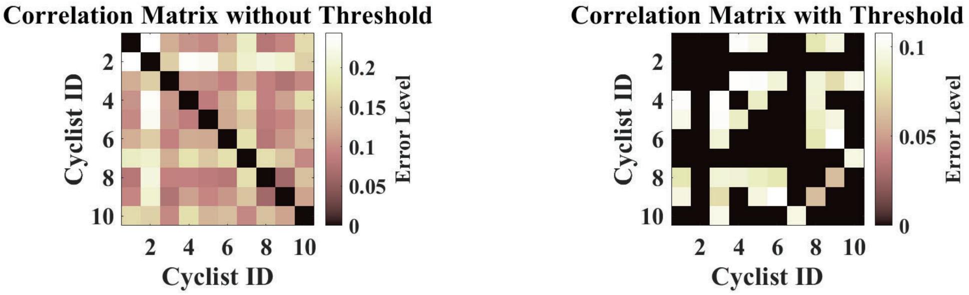

Generalized Hull-Space

Entity Specific Hull-Space

Figure 14: Observed motion space over all cyclists and for one representative example shape

The hull in Figure 14 strongly indicates the influences of the forced seating position with the two feet firmly on or at the pedals. At the same time, it can be read from the standardized width scale that a cyclist needs more space in the width dimension than the pedestrian, which is not only due to the nature of the handlebars, but also to perform driving related gestures like hand signals.

Results Grouping Cyclists
The left quasi-correlation matrix from Figure 15 already shows a nearly 50% lower maximum error level compared to the distribution of pedestrians from Figure 11.


Correlation Matrix without Threshold

Correlation Matrix with Threshold

Figure 15: Cyclist correlation matrix

The results from Figure 16 support the initial interpretation of the lower error level from the previous figure. There was no threshold at which more than one group with more than one group member would have formed. Therefore, for comparability, the same threshold was chosen as for the pedestrian. Along with this, one large group was formed, with only a single reconstruction not adhering to the group resemblance.

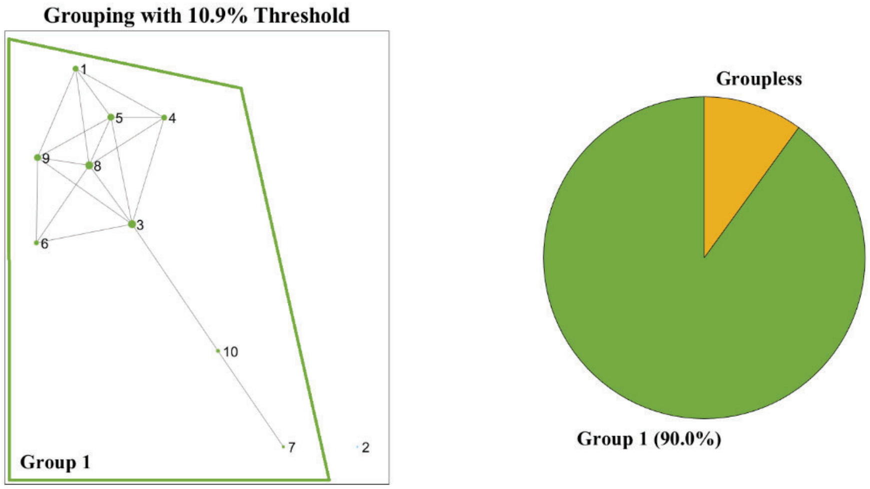


Figure 16: Resulting groups of cyclist behavior

Results Shape and Pose Cyclist

The mean shape from the grouped cyclist data is shown in Figure 17.

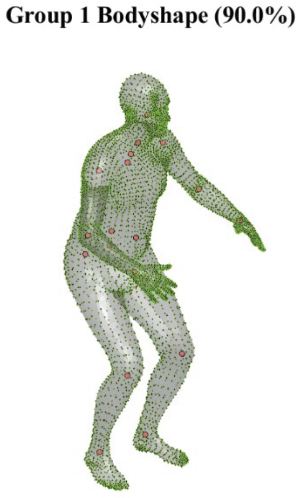


Figure 17: Mean cyclist pre-crash shape for group larger one subject

The shape subjectively matches that of a cyclist, but it projects legs and feet parallel to each other. This may have been caused by the applied averaging methods or by an error in the computer-vision-based shape reconstruction.

In general, there is no evidence of any special behavior of cyclists that differs from the natural driving behavior.

The joint angles of the singular mean cyclist position, as they are listed in Table 3, do not show any irregularities at all. Similar to the pedestrians, the cyclists' pose also withstands the test for biomechanical physiological plausibility. As such, they form a plausible and objectively tangible foundation for future model driven test case studies with cyclist VRUs.

Table 3: Cyclist mean pose joint angles

	Group 1		
JOINT ID	**X [°]**	**Y' [°]**	**Z'' [°]**
1 (L Hip)	9.79	-36.07	0.42
2 (R Hip)	-8.79	-37.82	-3.74
3 (Spine 1)	0.48	24.49	1.25
4 (L Knee)	-14.95	55.10	3.68
5 (R Knee)	14.63	58.41	-11.23
6 (Spine 2)	-0.06	-3.61	-1.05
7 (L Ankle)	2.92	-14.04	8.20
8 (R Ankle)	4.45	-13.93	-10.33
9 (Spine 3)	1.60	-1.29	1.62
12 (Neck)	-4.12	-5.04	6.12
15 (Head)	4.56	-3.47	2.24
16 (L Shoulder)	37.53	-34.33	-15.37
17 (R Shoulder)	-39.35	-34.09	17.28
18 (L Elbow)	21.49	-58.02	4.28
19 (R Elbow)	-19.40	-54.28	-7.64
20 (L Wrist)	-0.91	-8.58	0.97
21 (R Wrist)	-0.74	-8.86	-5.53

DISCUSSION

The work presented in this submission is subject to several limitations. The GIDAS database gathers in-depth information about accidents with casualties weighted for Germany, which might not strictly apply to the video database from South Korea. Moreover, the video data shows that such information is rather essential when it comes to the behavioral aspects of VRU during the pre-crash phase, as such information cannot be retrospectively gathered from on-site investigations. This in-depth data would help to further specify the mechanisms of injuries sustained and to reconstruct the vehicle speed. All in all, both data sources would allow for a wide range of accident crash causation analysis.

The video footage used in this study is characterized by a low sample rate and a low resolution. Some misinterpretation of the behavioral aspects might therefore occur during the kinematics estimation processes. Due to the low sample rate, components of a movement may also have been lost because the movement signal was under sampled. Additionally, there is no "ground truth" data available which could be used to assess the validity of the estimations. This point was addressed at least to the extent that the mean postures were tested for physiological plausibility at the joint angle level.

An additional uncertainty is introduced through the missing representation of the individual body segment lengths of every subject. The used shape model scales its volume but not the average length of each body segment. This leads to a constant length relation of upper- to lower leg or torso length to leg length. This is not the case in nature and might result in an additional error regarding the estimated joint location. Also, the used shape model is not scalable in its total height. This affects mainly the volume of the estimated hull which presently only represents a 50-percentile male. It would thus be necessary to scale the hull volume relative to the real body height of the pedestrian or cyclist involved in the accident. The additionally collected kinematic data is not affected by this issue since they are in general scaling invariant.

Nevertheless, the general shape estimation from frame data is still the best method to obtain plausible information from real live video material. As the quality of the underlying data may increases, so will the quality of the reconstruction in future applications. Applying the method to high resolution data may however not require additional preprocessing to improve the image quality. Still, the study shows that even with a low sampled video stream, a pose and motion database can be setup without necessitating high end video hardware.

The sample rate played only a minor role in the statistical grouping applied in this work. The authors have successfully found a method to classify different types of VRUs in a time-efficient and objective way without having to train additional neural networks in advance, which makes the method very transparent as well. Nevertheless, this method also offers potential for optimization and further development. One obvious weakness is that the evaluated pose similarity is based on only one frame. With a better sampling, however, the correlation of the kinematics over time would be much more informative. In addition, it is debatable whether all joints should be equally important for the similarity evaluation or whether they should be weighted according to a criterion such as the distance to the center of the body. Finally, this work shows that average poses for VRUs could be determined but their influence on the outcome of a crash is yet unclear. For the final differentiation of tests with post-mortem human subjects and passive models, it would now be necessary to determine what distinguishes the poses on a dynamic level and how they differ from each other on the force and momentum side. However, this could be addressed with muscle-driven human body models and the data we have identified as a target variable in future studies.

Applying the ATTENTION criteria to Germany, the study furthermore reveals that the priority should be set on car crashes against cyclists as these are represented with a share of 8% of all crashes with personal injuries.

As the sample shows, in nearly every 2nd crash (43%) the VRU did not recognize the oncoming vehicle. However, in case of perceiving the vehicle, different reaction patterns were observed. These patterns can be further evaluated using objective criteria such that a classification of the patterns is made possible. A similar behavioral pattern is also found in the relative group distribution across the pose grouping. Here, little to no visible reaction to the upcoming crash was found for at least 34.4%. Thus, the subjective and objective analyses come to a comparable conclusion in terms of their magnitude. In addition, 15.6% of all cases still were without inferred motion intention. It is not yet possible to say whether these are generally unusual behaviors, if their outer appearance was strongly influenced by an additional attraction or whether this is a phenomenon based on the small sample size.

With the cyclists on the other hand, the distribution of physical appearance was much less diverse, resulting in a single group instead of three. At the same time, the subjective evaluation of the behavior was much more difficult than with the pedestrians, which was also due to the reduced group size. Considering the result that there is no special precrash defense motion for cyclists, we conclude that the decisive influence for the outer shape of a cyclist is the type of bicycle which is being ridden. Thus, the only person who fell out of the presented grouping was riding a road bike where otherwise roadsters were present in the videos. In this context, the result represents a good and objective reference for those who are interested in the reproduction of natural movements and postures on a bicycle. Nevertheless, it can also be seen that cooperative crash avoidance by the VRUs cannot be assumed when it comes to the development of crash avoidance systems.

For both samples the determination of the impact angle was also rather limited as no objective criteria were applied on the video data. Nonetheless, the results give some valuable input when it comes to the impact itself as a crossing VRU will have different injury patterns compared to impacts in which the VRU is hit head-on.

The objective analysis methods presented in this work can help to improve behavior categorization. Generalized and scaling invariant joint angles can be further used for simulative situation analysis via pose and motion mimicking in a situation reconstruction. A typical approach to measure such kinematics is via marker-based motion capturing, which is not possible (for ethical reasons) for the relevant use case. We are therefore limited to marker-less tracing methods, that are applicable retrospectively to make the most of past recordings such as the available dashcam videos. Since these are only available as 2D data, but 3D information is mandatory to reconstruct the whole body's position and local joint angles, we are limited to methods that are using dimensional estimation strategies via AI to bypass the redundancy problem. One possible approach that was already used by Technical University of Graz (TU Graz) [4] for their behavioral categorization, is the application of a multidimensionally scalable human shape model which allows to reconstruct not just 3 translational but 3 rotational dimensions in the current use case as well.

CONCLUSIONS

The results of this study show that a pre-crash behavior of VRUs can be objectively determined from low-quality in-field video data using AI-driven methods and that it differs from regular human locomotion patterns. Furthermore, it shows that this video data can be used to set up a position and movement database. Both lay the foundation to estimate an injury risk index of VRUs in the following stages of the ATTENTION project.

State-of-the-art accident research uses data from post-accident investigations which is gathered in databases like GIDAS. Information about the impact moment may be inferred but the pre-crash behavior of VRUs is usually out of reach for these kinds of investigations. To the authors' knowledge, this paper presents the first systematic reconstruction of how people behave before the crash and react to the imminent impact, e.g., do they react at all or change their behaviors in distinct ways. This data can be used to supplement retrospective data which may lead to a better understanding of crash causes, injury patterns and new safety solutions from the automotive industry for protecting VRUs. Therefore, the dashcam video data is a valuable additional asset. Currently, the amount of video data analyzed is limited. To complete the picture of the behavioral patterns of VRUs prior to impact, more data should be added to form a comprehensive database.

Finally, and thanks to its automated and generalized nature, our method should be applied to a larger amount of dashcam video data to increase the statistical significance of this work. The great advantage of our method is that it is possible to objectively evaluate and compare large amounts of results in a short time.

Within the project ATTENTION, the behavioral pre-crash patterns of VRUs will be used as an input for crash simulations. The severity of the injuries sustained by the VRUs within these crash simulations will be determined and mapped to the behavioral patterns which will form a new database for crash analysis. The final goal within the project ATTENTION is to deduce the potential injury risk of a VRU before an accident, based on the detection of these, their behavior before the impact as well as the trajectory of the involved vehicle. This may give rise to new driving functions using an optimized combination of evasive steering and emergency braking to protect the VRU in an unavoidable accident as well as possible. This would provide further insight into which avoidance or mitigation strategies can be initiated in case of unavoidable VRU accidents.

ACKNOWLEDGEMENTS

The project "ATTENTION" (long title: Artificial Intelligence for real-time injury prediction) on which this report is based was funded by the German Federal Ministry of Economic Affairs and Climate Action (Bundesministerium für Wirtschaft und Klimaschutz) under the funding code 19A21027A.

REFERENCES

[1] Global status report on road safety 2018. Geneva: World Health Organization; 2018. Licence: CC BYNC-SA 3.0 IGO. ISBN 978-92-4-156568-4

[2] European New Car Assessment Programme (EuroNCAP) – Test Protocol AEB/LSS VRU systems, Implementation 2023, Version 4.3, November 2022, accessed on November 28th 2022 at www.euroncap.com/en/for-engineers/protocols/vulnerable-road-user-vru-protection/

[3] Liers, H.. 2018. "Traffic accident research in Germany and the German in-depth accident study (GIDAS)", SIAM Conference

[4] Schachner, M. and Schneider, B., and Klug, C. and Sinz, W.. 2020/9/1. "Extracting quantitative descriptions of pedestrian pre-crash postures from real-world accident videos", International Research Council on Biomechanics of Injury (IRCOBI).

[5] Sul, J. and Cho, S.. 2009. "Obtaining and applying of traffic accident data using automatic accident recording system in Korea". 4th International Road Traffic Accident Database (IRTAD) conference, Seoul. Korea. Pp.- 395-395

[6] Sulzberger, L. and Schmidt, D. and Moennich, J. and Schlender, T. and Lich, T. 2022. "How to use historic accident data for a reliable assessment of traffic safety measurements", 7th International Road Traffic Accident Database (IRTAD) conference, Lyon, France.

[7] Federal Statistical Office of Germany (editor). August 2021. Special evaluation of traffic accident 2020 on behalf of Bosch Corporation

[8] Federal Statistical Office of Germany (editor). 2021. "Verkehrsunfälle 2020, Fachserie 8, Reihe 7". www.destatis.de

[9] Lich, T. and Moennich, J. and Schmidt, D. and Voss, M.. 2022. "Preparation of an AI based real-time injury risk index estimation by deriving road user behavior from video-documented crashes", 15th International Symposium and Exhibition on Sophisticated Car Safety System (airbag2022), Mannheim. Germany. Fraunhofer ICT (editor). ISSN 0722-4087. Pp. 20.1-20.19.

[10] Wang, X. and Yie, L. and Dong, C. and Shan, Y. 2021. "Real-ESRGAN: Training Real-World Blind Super-Resolution With Pure Synthetic Data" In Proceedings of the IEEE/CVF International Conference on Computer Vision, 1905-1914

[11] Kocabas, M. and Huang, C.-H. and Hilliges, O. and Black, M. 2021. "PARE: Part Attention Regressor for 3D Human Body Estimation" In Proceedings of the IEEE/CVF International Conference on Computer Vision, 11127-11137

[12] Cao, Z. and Simon, T. and Wei, S.-E. and Sheikh, Y. 2017. "Realtime Multi-Person 2D Pose Estimation Using Part Affinity Fields" In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 7291-7299

[13] Loper, M. and Mahmood, N. and Romero, J. and Pons-Moll, G. and Black, M. 2015. "SMPL: a skinned multi-person linear model" ACM Transactions on Graphics 34, No. 6, 248:1-248:16

[14] Kolotouros, N. and Pavlakos, G. and Black, M. and Daniilidis, K. 2019. "Learning to Reconstruct 3D Human Pose and Shape via Model-Fitting in the Loop" In Proceedings of the IEEE/CVF International Conference on Computer Vision, 2252-2261

[15] Ionescu, C. and Papava, D. and Olaru, V. and Sminchisescu, C. 2014. "Human3.6M: Large Scale Datasets and Predictive Methods for 3D Human Sensing in Natural Environments" IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 36, No. 7

[16] Barber, C.B. and Dobkin, D.P. and Huhdanpaa, H. 1996. "The quickhull algorithm for convex hulls" ACM Transactions on Mathematical Software 22, No. 4, 469-483

[17] Markley, F.L. and Cheng, Y. and Crassidis, J.L. and Oshman, Y. 2007. "Averaging Quaternions" Journal of Guidance, Control, and Dynamics 30, No. 4, 1193-1197

[18] Barter, T. and Emmanuel, I. and Truett, B. 1957. "A statistical evaluation of joint range data" WADC Technical Note 53-311, Wright Patterson Airforce Base, OH

A METHOD FOR EFFICIENT GENERATION AND OPTIMIZATION OF SIMULATION-BASED TRAINING DATA FOR DATA-DRIVEN INJURY PREDICTION IN VRU-VEHICLE ACCIDENT SCENARIOS

Niranjan, Ballal
Thomas, Soot
Michael, Dlugosch
Niclas, Trube
Fraunhofer-Institute for High-Speed Dynamics, Ernst-Mach-Institut, EMI
Germany

Dirk, Fressmann
DYNAmore Gesellschaft für FEM Ingenieurdienstleistungen mbH
Germany

Paper Number 23-0215

ABSTRACT

Urban traffic is characterized by limited space, varying traffic flows and multiple types of road users. Despite increasing automation and design efforts, the joint use of traffic areas poses a particular risk for vulnerable road users (VRUs). In order to make traffic as safe as possible, the severity of injuries to VRUs in unavoidable collisions must be reduced. In future applications, predicting situation-specific injury risks for VRUs in real-time using machine learning (ML) could support decision making in determining risk minimization strategies. The predictive capability of any ML model is determined by the quality of the used training data. While there are no real-world training data available for injury prediction, simulation data, which is frequently employed in passive safety engineering, can be used as synthetic data. Since deliberate training data generation consumes substantial resources, particular attention is focused on the iterative generation of optimized simulation data sets. This study presents and discusses an adaptive simulation data generation pipeline to generate simulation data sets that reflect the overall system's behavior with the overall goal of efficiency and sustainability.

The novel pipeline involving nine steps is divided into two phases, "Data Generation" and "Data Exploitation". The "Data Generation" phase predominately focusses on the adaptive strategies to generate a generalist training data set. Along with the fundamental techniques for adaptively adding new points, metrics for assessing the information content of the present data set and for tracking the iterative sampling progress are also discussed in this study. Additionally, experiments to understand the effects of batch size is conducted and the potential use of information content metrics for process termination and dynamic, adaptive batch size adjustment is discussed. The pipeline is initially tested using a generic example and is then applied to a simulation setup modeling a human head crashing onto a vehicle windshield. The observations from applying the pipeline to the simulation setup are compared with the observations from applying it to the generic function to evaluate the novel pipeline.

It is shown, that the pipeline is generally applicable to such real-world problems and that the anticipated dynamic behavior of the data generation process is confirmed in the generic and real application example. This lays fundamental groundwork which needs to be extended along multiple routes in future work.

MOTIVATION

Not only in recent years, urban traffic systems have shown a significant trend towards multimodality [1]. Next to motivating an extensive amount of research activities on the design of multimodal urban mobility systems, this poses significant challenges towards traffic safety to all stakeholders involved [2] [3]. Particularly vulnerable road users (VRUs), e.g. pedestrians or cyclists, exhibit an overproportionate share, an increased injury severity and relatively high death-rates in the accident data [4] [5] [6]. In order to respond to these VRU-specific needs in traffic safety, a consortium of industrial and academic partners has teamed up in the research project ATTENTION ("artificial intelligence for real-time injury prediction"*) to develop a framework, as well as constituting methods and tools to dynamically predict injury risks for VRUs in accident scenarios [7]. Given the requirement of (near) real-time prediction, conventional engineering methods to predict the behavior or performance of structures under dynamic loading conditions – namely the finite element method (FEM) – are not feasible. More precisely, while the comprehensive simulation of a crash scenario can take up to 30 h on an advanced compute cluster, the collision of a vehicle is avoidable until ca. 1.5 seconds before impact [8]. Hence, the potentials and applicability of artificial intelligence (AI) or – more accurately – machine learning (ML) to predict the system's responses in such scenarios are studied in this project.

ML has shown to produce promising results in predicting the behavior of vehicle structures under crash in several studies [9] [10] [11]. As outlined by Kohar et al. [10], one of the main challenges in applying ML in engineering design for crashworthiness lies in the limited availability and heterogeneity of suitable training data. Compared to other domains, the mechanical engineering domain is generally dealing with highly complex systems and challenges, which are tackled using advanced modeling and simulation methods (such as FEM) requiring substantial (computational) resources. This constitutes two distinct characteristics of this domain hindering the widespread adoption of ML-based methods: data scarcity and data complexity. As advocated by Ng [12], shifting the ML paradigm from model-centric to data-centric approaches is an effective way to efficiently increase the overall performance in a wide range of ML applications. As opposed to rather conventional model-centric approaches, data-centric ML doesn't focus on engineering the ML-model itself (e.g. model type and architecture) to increase the overall performance, but on engineering the data used to train the model [13]. In view of aforementioned domain characteristics, this fundamental notion, that data quality has a stronger impact on the performance, efficiency and scalability of ML solutions than model sophistication or fidelity, particularly applies to the engineering domain. In order to cope with the resource-capacity induced limited availability and complexity of simulation data, following a data-centric approach in these types of applications can be considered a logical consequence. While leveraging legacy simulation data as training data poses additional – despite interesting approaches (see e.g. Vasu et al. [14] or Greve and Van de Weg [15]) widely unsolved – challenges, deliberately generating sets of training data through simulation for a specific prediction task is a common practice in respective current R&D activities [16] [17]. In the majority of applications, the training data set is initially generated using well established design-of-experiments (DoE) methods to sample the design space and FEM-simulation software. With that, an ML model is trained to predict the system response parameters of interest based on the input parameter settings as features. Since these unidirectional "one-shot" data generation approaches do not allow for any feedback from the training and model performance to the sampling phase, designing for overall process efficiency is inhibited. As seen in the works of Chec [18] and Kayvantash [19], modifying this pipeline to resemble an active learning (AL) scheme is one approach to introduce this feedback loop and thus maximize the model performance (e.g. accuracy) while minimizing the number of (often expensive) samples in the training data set. Here, by iteratively generating batches of data and evaluating the current performance of the ML model, the sampling locations for the next iteration are determined by maximizing their contribution to the learning process [20]. While these approaches have shown to yield an increased efficiency, one key shortcoming does persist in all of the existing methodologies.

As stated above, efficiently using resources (e.g. simulation) and managing all the digital assets related to the ML pipeline sustainably is always beneficial, but mandatory in the engineering domain. Hence, generating a data set for the purpose to train solely one specific ML model making one specific set of predictions – as it is also the case in regular AL schemes – does work for methods development in R&D but doesn't fulfil the domain application requirements. Rather than focusing on the ML model and its predictive capabilities while adaptively generating the training data, focusing on the data itself and optimizing the representation of the system's complex behavior in the data set holds significant potential. The goal is to generate a data set, which – within reasonable limits – represents all the relevant characteristic features of the system's (e.g. crash structure) behavior without tailoring it to fit a very specific ML application – and thus make it reusable in multiple applications. Combined with already partially employed transfer learning approaches [21] [22], this aims to increase the overall efficiency and sustainability by enabling the reuse of not only the generated data sets, but also the model(s) trained. For that, an adaptive data generation pipeline is proposed to efficiently generate information-dense and reusable data sets for training transferable ML models for traffic and vehicle safety applications. The pipeline is applied to the critical use-case of VRU safety.

INTRODUCTION AND STATE OF THE ART

As outlined above, applying ML in the domain of vehicle passive safety is an active field of current R&D efforts which have already produced quite promising results. In this chapter, two of the main ML pipeline architectures and selected fundamental methods are introduced and discussed with a particular focus on the respective data generation and sampling schemes.

Figure 1 shows the schematic flowchart of a typical ML pipeline used for data-driven predictions of system response parameters which could be constituting the crashworthiness of vehicle structures or the behavior of a dummy or human body model. While rounded rectangular shapes depict (sub-) processes in the pipeline, hexagonal and diamond shapes depict resources and process bifurcations, respectively. In general, the pipeline is divided into two major sections: the data generation and the data exploitation phase. After setting its dimensions and ranges, the DoE strategy to sample the design space is implemented in step 1. Given its comparably favorable characteristics (e.g. space filling) one widely adopted DoE-method is Latin Hypercube Sampling (LHS) [23] [24]. After defining the sample points – each of which could represent a certain crash scenario within the design space limits – the output responses are computed employing the respective FE model(s) and the compute resources (e.g. cluster) in step 2.

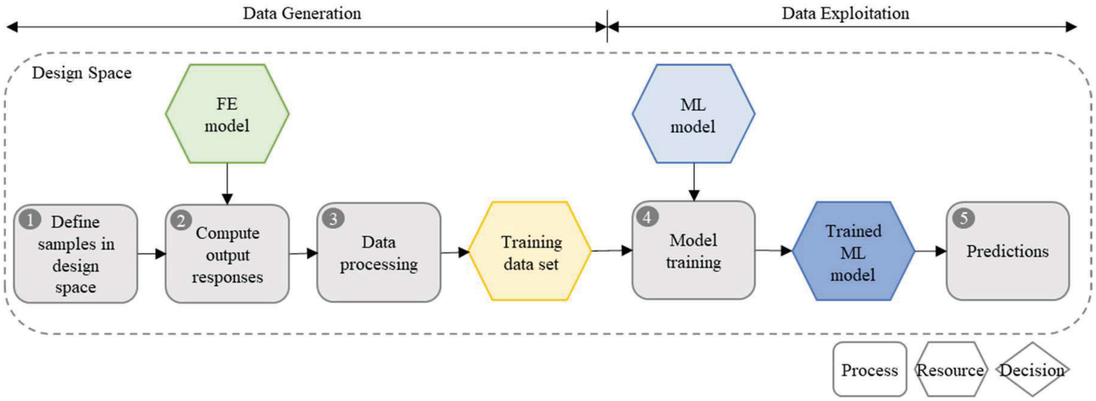


Figure 1. Generic flowchart of a conventional "one-shot sampling" ML pipeline.

Data processing in step 3 resembles conventional post-processing procedures such as the extraction and processing (e.g. re-sampling, filtering, structuring etc.) of relevant output responses from the simulation output files to generate suitable data for training purposes. The input parameters defining the sampled data points then are considered the features and the processed output responses (e.g. maximum acceleration at a defined location) are considered the labels of a resulting training data set for a supervised learning scheme [25]. In the vehicle safety domain, multiple feasible ML model types need to be assessed regarding their respective advantages and drawbacks in the specific application. For instance, while regular feed-forward neural networks (NNs) or random forests (RFs) can be used to predict discrete response values, gated recurrent units (GRUs) or long short-term memory networks (LSTMs) – both types of recurrent neural network (RNN) architectures – serve to predict time series, such as acceleration curves or node trajectories, accounting for the dependencies along the time series data [26] [27]. Potential benefits and drawbacks to consider might be model fidelity (and the corresponding amount of needed training data and computation time) or the interpretability and explainability of the model, which are key characteristics, particularly in safety critical applications [28] [29]. The training process in step 4 provides the generated training data set to the selected model reserving subsets for testing the model performance following a test-train-split or a cross-validation scheme and final validation with unseen data [25]. The trained model can then be used to predict output responses from input features within the limits of the design space in step 5.
Given this "one-shot-sampling" pipeline architecture, the information content of the training data set solely depends on the initial sampling in step 1. The linearity of this architecture does not allow for any internal feedback from data exploitation to the data generation phase, resulting in the fact that engineering the sampling methods in step 1 based on domain knowledge, experience and/or learnings from previous linear pipeline executions are the only – albeit inefficient – means to generate information dense, ideally suitable training data sets.

More advanced approaches towards training data generation are found in the field of active learning methods. Active learning – also called "query learning" – is a subfield of ML and follows the key hypothesis, that training efficiency is maximized, when the ML model can dynamically "query" additional data samples considering the current training progress [20]. Hence, through dynamically adding specific samples, the training data set continuously adapts to the information needs in the training process. Figure 2 depicts such an adaptive sampling scheme in the application scenario under consideration here.

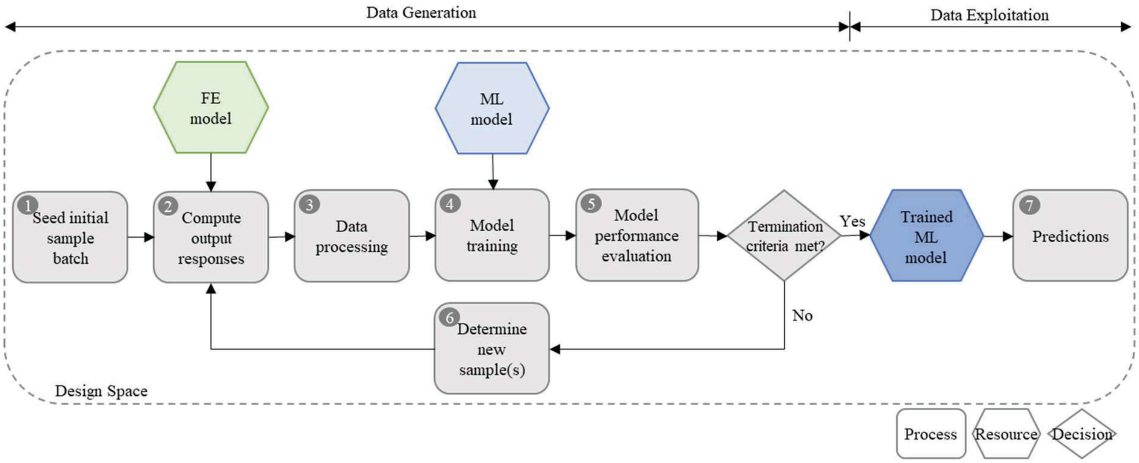


Figure 2. Generic flowchart of an adaptive sampling scheme in an AL pipeline.

Apart from potential effects on the overall process from the size of the initially sampled batch in step 1, the first four steps do not differ from the “one-shot-sampling” scheme in Figure 1. After that however, the prediction performance of the current (training) state of the model is evaluated considering the respective criteria (e.g. prediction accuracy based on error metrics such as the root-mean-squared-error (RMSE) with respect to the test samples [30]). A pre-defined threshold for the model performance can be used as a termination criterion for the data generation phase, which can be complemented by additional criteria such as a maximum number of samples in the training data set. As long as none of these termination criteria is met, new sample points to be added to the data set are determined in step 6. In AL many strategies for selecting additional samples – such as “rapid exploration”, “maximized model change” or “reduction of estimated error” have been established [31] [32]. According to Settles [20] and Géron [30] one of the most popular strategies is “uncertainty sampling”, where the additional samples are added in areas of the highest uncertainty of the current model, which is particularly interesting in combination with Gaussian Process (GP) models, since they indicate model uncertainty based on the variance (see also next chapter) [33] [34]. The data generation phase with adaptive sampling is terminated when the respective conditions (e.g. model prediction performance) are met. The trained model can then be used to make predictions within the design space in step 7.

Several examples in the engineering and materials science domain demonstrated the general applicability and potential benefits of such AL schemes [18] [19] and have generally shown that deliberate adaptive sampling strategies are superior to random selection in minimizing the number of experiments needed [35]. However, considering the dynamics inherent to such a pipeline architecture, the specific tailoring of the generated training data set for the model and prediction task at hand is an obvious consequence. This implies, that neither the model nor the data set are predestined to be reused in similar application scenarios in a sustainable way. While using performance metrics (e.g. accuracy) of the model to monitor and steer the data generation and training process enables optimizing for the specific prediction tasks, it also bears significant risks for blind spots regarding the representative quality of the data set.

THE ADAPTIVE DATA GENERATION PIPELINE

Pipeline overview

Building on the previously introduced and established data generation and learning schemes, a novel general pipeline architecture is proposed. With the intention to efficiently generate reusable training data sets representing the definitive characteristics of a system’s behavior (here: crashworthiness of structures under crash conditions) this general architecture allows for its customization to the respective application at the most critical steps. As depicted in Figure 3, the novel pipeline consists of the two phases “Data Generation” and “Data Exploitation” as well. However, one of the main distinctive features of this architecture is that the result of the adaptive data generation phase is the generalist training data set instead of a trained ML model. Although this phase closely resembles the one from the AL scheme, there are some key differences to be pointed out. In the first iteration, after processing the data in step 3, a subset of the data is branched off to serve as an unbiased test set for later training processes in the second phase. (Depending on the overall number of samples and the size of the seed batch this might also occur in another early iteration). This is relevant, since the adaptive sampling process can also be controlled using formalized expert knowledge, which is expected to introduce a (beneficial) bias into the data set leading to more representative characteristics but does not reflect the probability distribution of the application scenario. This could also be interpreted as a “data set overfit”. Steps 4 and 5 serve to analyze the current data set for its information content and to monitor the progress of the iterative sampling loop. It is crucial to state that the one - or multiple - ML models trained in step 4 merely serve as tools to probe the data set in generation and are not used in later prediction tasks. In order to reach the goal of a generalist training data set, the response quantity predicted by the ML models during the data generation phase could be different from the response quantities considered in the later application (data exploitation phase). By combining several output quantities of the system, one could in addition introduce expert knowledge in order to evaluate the information density of the data set in a generalized way. The data quality evaluation is based on the predictive qualities developed by the trained models but can (and should) be extended by additional metrics such as importance-driven sampling density in individual dimensions or "regions" of the design space. Steps 4 and 5 is where the representative capabilities of the data set are optimized using customized metrics based on formalized expert knowledge, which, for instance, could be stated as "(relative) information density requirements".

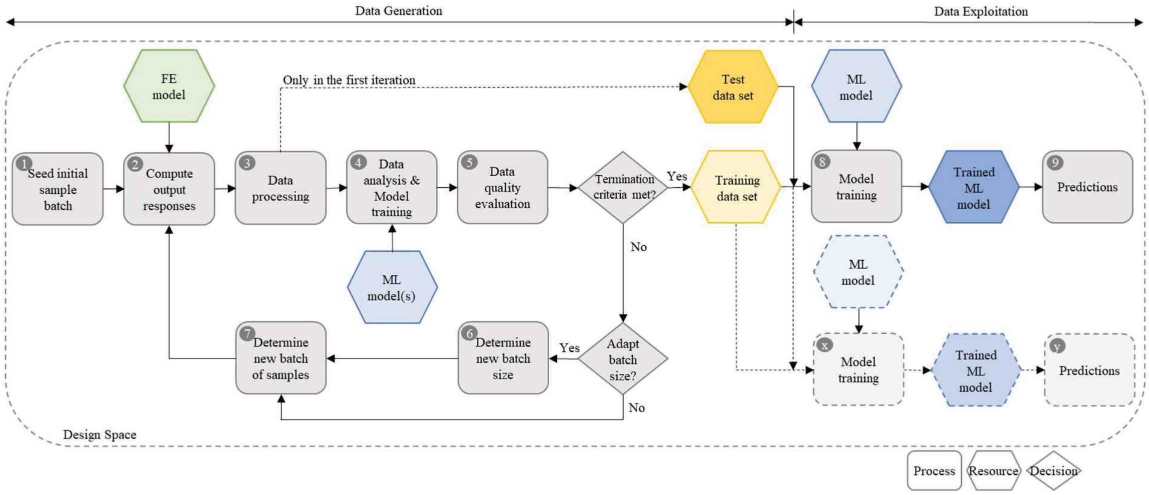


Figure 3. Generic flowchart of the proposed adaptive data generation pipeline.

After step 5 the first check taking place is for the fulfillment of any termination criteria. These could again be stated as a maximum number of samples or iterations. Additional criteria should refer to the data set quality and could be derived from model prediction quality metrics, data / information density analytics or – potentially even more powerful - gradients thereof basically representing metrics of convergence. Given that no sufficient criterion is met, these gradients can then be used in the next step checking the conditions for the adaption of the batch size (defining the number of samples generated and processed in the next iteration). These – relative and thus generalizable - conditions can be adapted individually for each application since they implement an optimum strategy to dynamically balance the resources spent for generating new samples (with expensive FE simulations) and model training or data analytics considering the current state of information content and its convergence, respectively. In step 6 the batch size is adapted to the latest gradient of the metrics used to estimate the convergence considering the predefined strategy. The new samples defined in step 7 are then selected based on the chosen sampling strategies (see previous chapter). Depending on the models and analytics employed, a promising combination of strategies might be "uncertainty sampling" with an additional "exploration" or "space filling" criterion [36]. These samples, or rather the respective system responses, are then computed back in step 2.

In Figure 3 the second phase of "Data Exploitation" schematically depicts the multiple paths of training (step 8 or x) ML models and using them to make predictions (step 9 or y) within the limits of the design space using the generated data set. This is generally enabled, since the data set is engineered to represent the systems' characteristic behavior, rather than being tailored to specifically fit a certain model and prediction task (e.g. predict structural kinematics - not critical values for specific crashworthiness performance metrics). Naturally, there is a trade-off between generalizability and respective specific prediction quality, which is expected to also be specific to the individual system and task(s). However, especially given the particular conditions in vehicle safety engineering design, there are at least three potential scenarios benefitting from such an overall approach. First, it can be expected, that rather simple – and from an expert point of view - standard prediction tasks will be trainable with the same generated data set. And that applies to a comparably larger share of the design space than with existing pipelines with a high degree of specialization. Second, by employing transfer learning schemes [37], it is generally possible to re-train a model to perform in a similar prediction task by adding few data points to the training data set, which will increase the overall process efficiency and sustainability. Third, similar to the transfer learning approach, it can reasonably be anticipated that a data set generated with the proposed pipeline and a respective set of metrics will function as a baseline or "fundamental" data set, which comprises the majority of the system's relevant information. This baseline data set can then be extended with only a few additional sample points to efficiently customize it as a branch for a specific application. This could also happen adaptively, which could then resemble a combination of phase 1 of Figure 3 with the AL scheme depicted in Figure 2, where the initial seed batch (step 1) would be the adaptively generated baseline data set.

Detailed description of the core processes and algorithms in the current implementation

Following the conceptual introduction of the proposed pipeline architecture, this chapter describes a first base implementation and the respective algorithms used in the individual steps. As preliminary note, it is stated that this implementation is intended to study the dynamics and impact of the overall approach and can be considered the groundwork for multiple enhancements, extensions and complements in the future. In the following, all steps or sub-processes are briefly described with their algorithms and the underlying theory.

Step 1: The initial samples are generated using Latin Hypercube Sampling (LHS), which is a well-established and widely used sampling method in the engineering community. The core idea is to divide the design space into “boxes” (or hypercubes) of equal probability and sample one data point randomly within each box. This method is mainly used for relatively sparse sampling schemes and provides rather stable results (with respect to mean value and distribution) compared to other methods such as Monte Carlo Sampling. [24]. To even improve the space filling characteristics of LHS one could implement Optimal Latin Hypercube Sampling (OLHS) which enforces a certain minimum distance between samples, but might increase computational costs significantly – especially for high-dimensional design spaces [38].

Step 2: In engineering structural design applications such as the one at hand, the system’s responses are computed using a regular FE simulation framework such as LS-Dyna [39]. In general, these “ground truth” observations might however also be generated by evaluating an analytic function or any other “oracle” as termed in the AL domain.

Step 3: With FE simulation frameworks the system’s response is often written to a binary file, which is generally incomprehensible to humans. Python modules such as lasso-python [40] aid in reading, writing and automated processing of these binary files. Additionally, because these files are gigabyte-sized, it is efficient to read the necessary response parameters for multiple simulations and tabulate them using python modules such as pandas [41]. In this implementation, the entire process from multiple raw simulation data output to easily processable, aggregated and tabulated parameters of interest is automated with python scripts and pandas dataframes.

Step 4: In the current implementation, the ML model used is a Gaussian Process (GP) regression model. The GP represents a generalization of the Gaussian distribution and can be used to define distributions over functions [42]. A GP is defined by its mean and covariance function parametrized using hyperparameters and can be utilized as prior for Bayesian inference [42]. The posterior mean and variance are determined considering the training data and can be used to make predictions at unseen locations of the design space [42]. In this study, the python based Gaussian Process framework GPy is used [43]. The applied kernel is based on a combination of *matern*, white noise and linear kernel considering anisotropic length scale parameters. One key benefit of using a GP model is that the model indicates variances, which can be interpreted as “model uncertainties”. This information can be used to significantly benefit strategic sampling of additional data points. Generally, all other model types are feasible to serve as data set probing tool in step 4. Running different models in parallel might again increase computational costs, but yield a rich assessment of the current data quality.

Step 5: Two distinct metrics are used to estimate the increase in information (density) and global convergence. The distance metric, here the root mean squared difference (RMSD), measures the change of the characteristics of the meta model response surface which is assumed to approximately quantify the relative information gain in the latest iteration – by comparing the model predictions at pre-defined evaluation points in the current iteration *(iter)* to the previous iteration *(iter-1)* using Equation (1).

$$RMSD = \sqrt{\frac{\sum_{i=1}^{N}\left(\hat{y}_{\ iter}(i) - \hat{y}_{\ iter-1}(i)\right)^{2}}{N}} \qquad \textbf{\textit{Equation (1)}}$$

$where$

N : $number\ of\ evaluation\ points$

$\hat{y}_{iter}(i)$: $predicted\ value\ of\ i^{th}\ evaluation\ point\ in\ the\ current\ iteration$

$\hat{y}_{iter-1}(i)$: $predicted\ value\ of\ i^{th}\ evaluation\ point\ in\ the\ previous\ iteration$

The performance metric, here the root mean squared error (RMSE), computes the standard deviation of the residuals or prediction error using Equation (2) to assess the improvement with respect to the actual (ground truth) value [44].

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}\left(y(i) - \hat{y}(i)\right)^2}{N}}$$

Equation (2)

$where$

N : $number\ of\ evaluation\ points$

$y(i)$: $actual\ value\ at\ i^{th}\ evaluation\ point$

$\hat{y}(i)$: $predicted\ value\ at\ i^{th}\ evaluation\ point$

In a real application, the RMSE is typically calculated on the data points that are not utilized to train the ML model. Due to the possibility of a sparsely sampled initial seed, dividing the dataset into training and evaluation datasets may result in evaluation points that do not cover the interesting areas of the design space. Cross-validation-based methods, on the other hand, can facilitate the ability to use all the available samples for the evaluation [45].

In the engineering design application of this study, where the ground truth is not available, cross-validation for RMSE calculation is utilized. The RMSD is calculated by evaluating the GP regression model at pre-defined high-density evaluation points. These could either be defined by densely grid-sampling the entire design space or by using another LHS step adding randomness to the selection process.

The current termination criterion is merely based on the maximum number of samples to be generated in the iterative process. This suffices to analyze the essential pipeline dynamics, but should be extended to thresholds for more sophisticated data set quality metrics and especially their gradients indicating information saturation.

Step 6: In this first implementation the dynamic batch size adaption is yet left to included. The following application examples do however study the effect of different – despite static – batch sizes on the overall pipeline dynamics. The findings can indicate potential strategies in dynamic batch size adaption considering information content convergence and resource needs at the different sub-processes.

Step 7: As mentioned in Step 4, one key benefit of using a GP model is that the model indicates variances, which can be interpreted as "model uncertainties." In the regions with lower variance, it can be assumed that the model uncertainties are lower. Therefore, the new samples are selected in those regions with the highest uncertainties to minimize the variances and thus increase the anticipated model prediction quality or “confidence” [46]. In order to locate new samples in the design space this is selected as the primary criterion, mathematically represented in Equation (3).

$$x_{new}^{\text{1st crit.}} = \underset{x \in D_{cand}}{\operatorname{argmax}}\, \widehat{\sigma}^2(x)$$

Equation (3)

$where$

x_{new} : $New\ sample\ point\ (location)$

$\widehat{\sigma}^2(x)$: $Model\ prediction\ variance\ at\ point\ x$

D_{cand} : $Predefined\ sample\ candidates\ in\ the\ design\ space\ D$

As it is stated above, Equation (3) is directly applicable only to the batch size of 1. When simultaneously adding multiple samples (e.g. batch size of *n*), the *n* points of largest model variance will be selected as next sample locations. With the progression of iterations and with the introduction of larger batch sizes, a pure variance-based addition of samples could result in a strong localization of samples in the design space. In order to prevent this local aggregation, a secondary distance-based criterion using a space-filling metric is utilized “intra-iterations” to prevent local clustering of the samples added within one batch and “inter-iterations” to prevent clustering of the samples added in the current and in previous iterations [36]. The space-filling metric S as mathematically represented in Equation (4) and the new sample selection in the design space using primary and secondary criterion as represented in Equation (5) are based on the works of Aute et al. [36].

$$S = 0.5 \times \max(S_N) \qquad \textbf{\textit{Equation (4)}}$$

$where$

$S :\ Space\ filling\ metric$
$S_N :\ List\ of\ Euclidian\ distances\ for\ all\ existing\ points\ in\ D_{train}$ to respective closest neighbor

$D_{train}: Set\ of\ existing\ training\ points\ in\ the\ design\ space\ D$

$$x_{new}^{\text{1st \& 2nd crit.}} = \underset{x \in D_{cand}}{\text{argmax}}\, \hat{\sigma}^2(x) \qquad \textbf{\textit{Equation (5)}}$$
$$\text{s.t. } \|x_{new} - x_k\|_2 \geq \text{S}, \qquad \forall\, x_k \in D_{train}$$

$where$
$x_{new} :\ New\ sample\ point\ (location)$

In this initial implementation the “Data Exploitation” phase not yet included. A full implementation can be achieved by extending the current one with a basic ML pipeline suitable to majority of applications in structural engineering. While this groundwork implementation suffices to reach the goal of this study, which is to understand the basic effects and dynamics of this adaptive data generation scheme while generating the data, it is clear, that an objective measurement of the overall performance (final prediction quality and overall data generation efficiency) is still left to be conducted in future work. A respective proposition is to be found in the last chapter of this paper.

Application of the pipeline to a generic example

As a first step, the data generation pipeline is applied to a generic mathematical example problem. This helps to demonstrate the functionality and dynamics of the pipeline while relating to rather clear expectations of the outcomes and having global access to ground truth.
Since it used in a wide range of meta-modeling and sampling methods applications, the function of choice for this study is the bimodal non-linear Hosaki function as defined in Equation (6) [47].

$$f(x) = \left(1 - 8x_1 + 7x_1^2 - \frac{7}{3}x_1^3 + \frac{1}{4}x_1^4\right) x_2^2 e^{-x_2} \qquad \textbf{\textit{Equation (6)}}$$

$where$

$$0.5 \leq x_1 \leq 4.5\ \ and\ 0.5 \leq x_2 \leq 4.5$$

In order to provide the reader with a clear image of the shape, Figure 4 depicts the response surface (*f(x)*) over the given value ranges from 0.5 to 4.5 for x_1 and x_2, respectively.

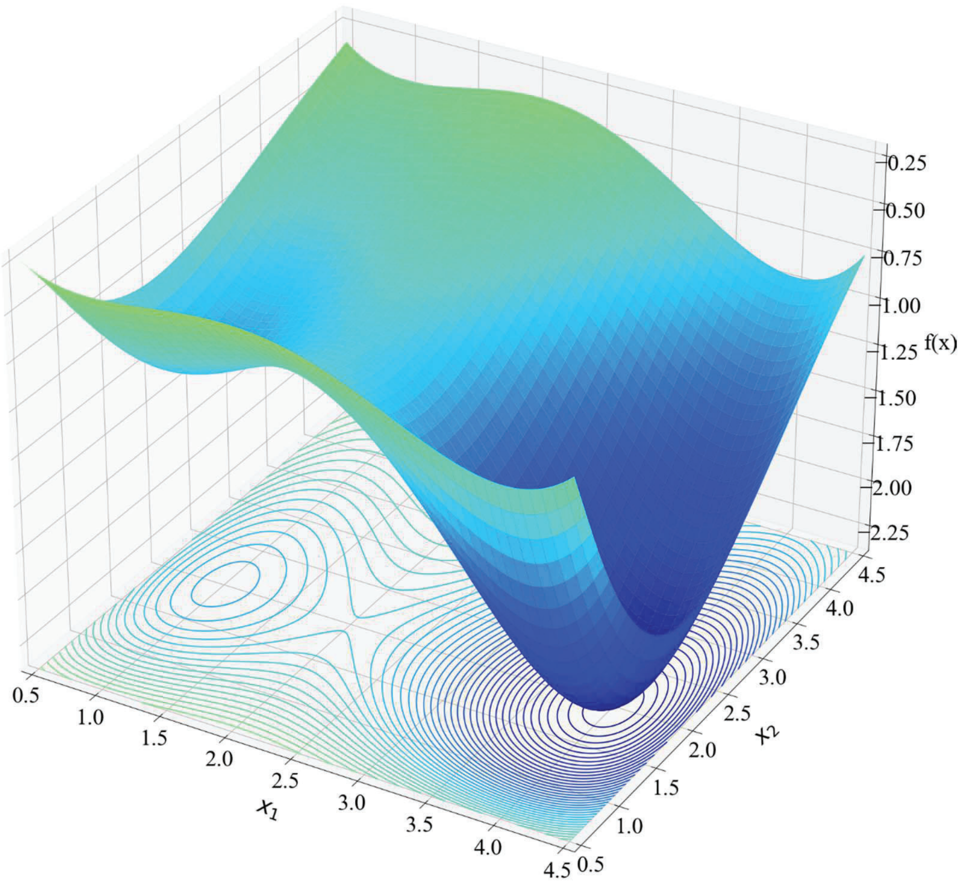


Figure 4. 3D plot of the Hosaki test function.

As for the boundary conditions of the experiment, it is stated that the evaluation points were defined by grid-sampling the design space with a total number 1000 equidistant points. The initial seed batch size was 5 and the maximum number of created samples was set to be 100. In order to study the effect of differently – despite statically – sized batches several experiments with fixed batch sizes of 1, 2, 4, 8 and ten samples per batch were conducted. The RMSE was calculated by relating the model prediction to the analytical ground truth given by Equation (6).

Figure 5 depicts the plot of RMSD and RMSE values over the generation of 100 samples (or iterations with a batch size of 1). For the RMSD and RMSE plots, a global convergence is observed as the model approximates the Hosaki ground truth with an increasing amount of information (samples) to be trained on. Even though a global convergence is anticipated overall, the early phase of sampling is also expected to see temporary increases in RMSE as a result of a sample point adding information to the very small body of existing information. This temporary increase is only observed at an initial stage and is not observed at a later stage, as would also be expected, when the body of existing information is already quite substantial. An additional effect might be that the model could, by coincidence, initially have seen critical, definitive samples. This average of high information quality is then drastically decreased by adding a sample of significantly lower learning value in an early phase.

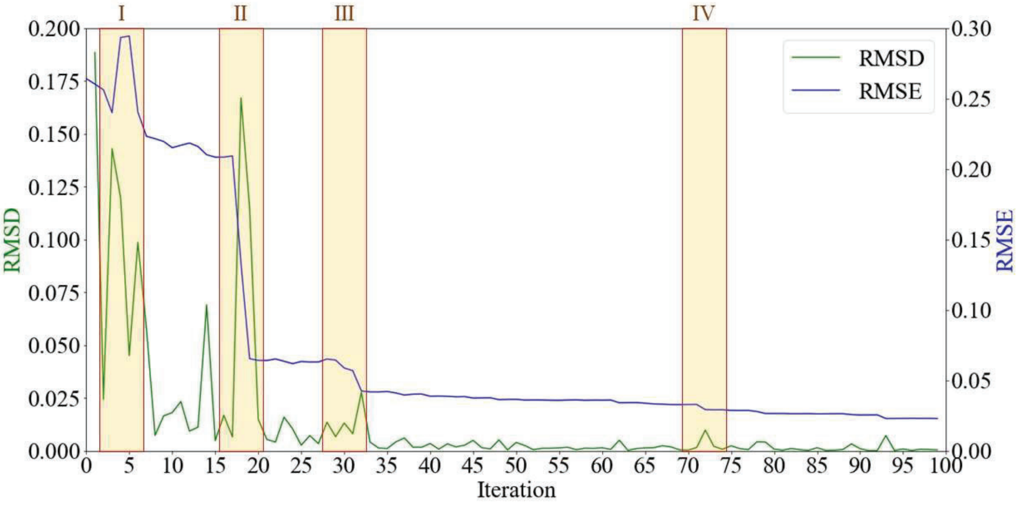


Figure 5. Plot of the RMSD and RMSE values over generated samples / iterations with batch size 1 for the Hosaki test function. Orange Boxes indicate the iteration ranges shown in detailed contour plots in Figure 6.

Along with model performance, it is seen that distinct features appear simultaneously in the RMSE and RMSD plots. This confirms the expectation that measuring a spike in RMSD, representing a significant change in the characteristic shape of the response, clearly relates to a significant change (for the better or the worse) in the RMSE. At a later stage, only decreasing RMSE values are anticipated with a “better educated” model and iterations 18 - 20 offers a remarkable illustration of the same. These peaks in RMSD are not limited to earlier stages, but are also identifiable at later stages in the sampling process. Despite the variations in RMSE being hardly noticeable when the model has already converged to a larger degree, the respective spikes in RMSD are comparably significant. This indicates their suitability to be used as termination criteria indicating information saturation.

Figure 6 depicts the contour plots for the model response surface, the residuum with respect to the Hosaki ground truth and the model variance for a selection of four times five consecutive iterations with batch size 1. These iteration streaks are also indicated in Figure 5.

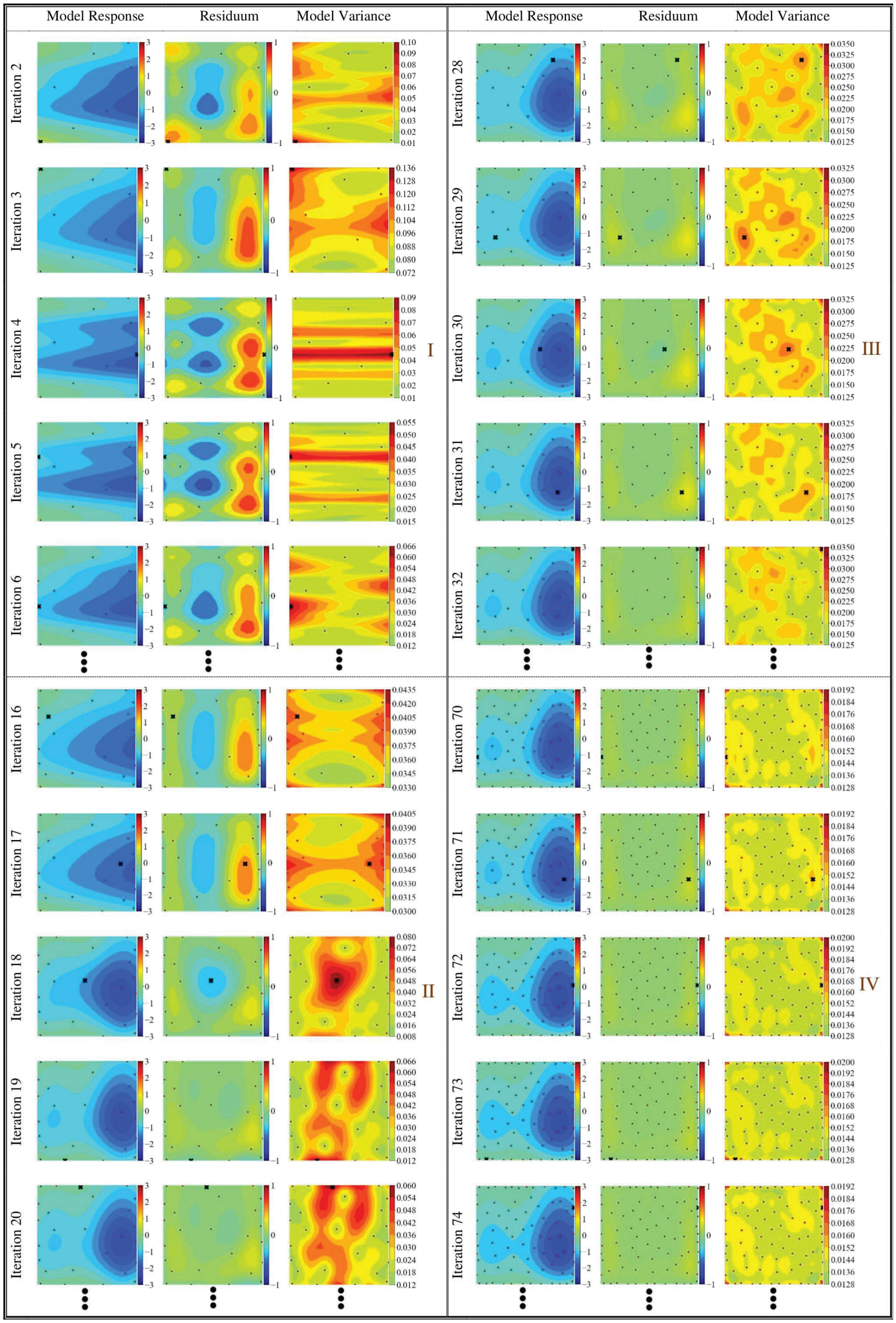


Figure 6. Contour plots of the model response surface (left), the residuum w.r.t. Hosaki ground truth (center) and model variance (right) for selected iterations (also indicated in Figure 5). The red and black dots indicate all currently available samples and the location of the new sample which is added to the training data to train the model of the next iteration (batch size 1), respectively.

As depicted in Figure 6, with increasing amounts of training information, the response surface convergence towards the Hosaki function is observed as expected. Simultaneously the residuum converges towards zero and does not show major changes in later phases. Additionally, the new point selection can be visually verified. The new sample points are always located where the highest model variance is indicated. In the following iterations this indication is shifted to a location without previously sampled information. Although the enforcement of the distance-based secondary criterion (especially the "intra-iterations") can't be observed here, it has diligently been evaluated and found to be effective.

Figure 7 depicts the plot of the RMSE over generated samples with multiple fixed batch sizes. Initially, it is seen that the resolution of the individual curves over the abscissa is defined by the batch size. This could be interpreted as a "model-wise" (as trained model) resolution of the respective performance in comparison to the models trained with a different batch size and thus different frequency.

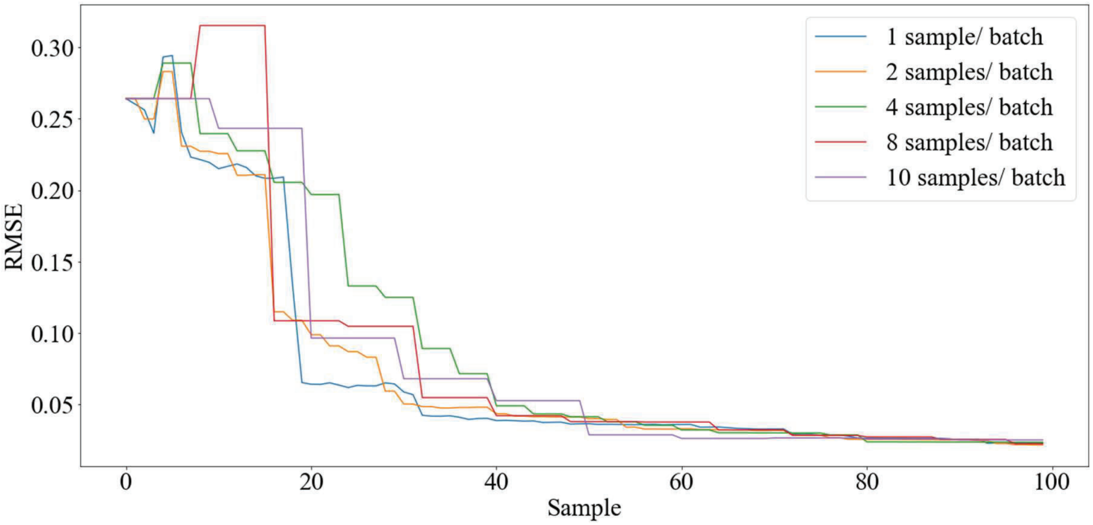


Figure 7. Plot of the RMSE over generated samples with multiple fixed (static) batch sizes for the Hosaki test function.

It can be observed that smaller batch sizes, as compared to larger ones, yield a gradual and a faster convergence towards lower error values. Particularly when looking at the higher plateau for batch size 8 over samples 8 – 16, one can conclude that adding a lot of information (larger batch) based on little (uncertain) information in an early stage does not result in a beneficial strategy regarding dynamic batch size adaption. The mutual convergence of the RMSE curves at a later stage indicates that the batch size doesn't affect the overall model convergence significantly. However, there are several other aspects (e.g. sub-process-specific resource consumption) to be considered when crafting a batch size adaption strategy.

Figure 8 depicts the plot of the RMSD value normalized by batch size over the sample generation for multiple fixed batch sizes. It can be observed that lower batch sizes yield higher values than larger batch sizes over the entire data generation process. This is expected since the information taken into consideration when defining a new sample point is maximized when minimizing the batch size. With a batch size of one, every single sample is the actual "next best sample" with respect to the (theoretically) available information. Increasing the batch size changes this ratio for the worse. This effect is additionally amplified by normalizing the "added value" per sample by batch size. Lower relative values per sample for higher batch sizes are then additionally related to a larger batch size number.

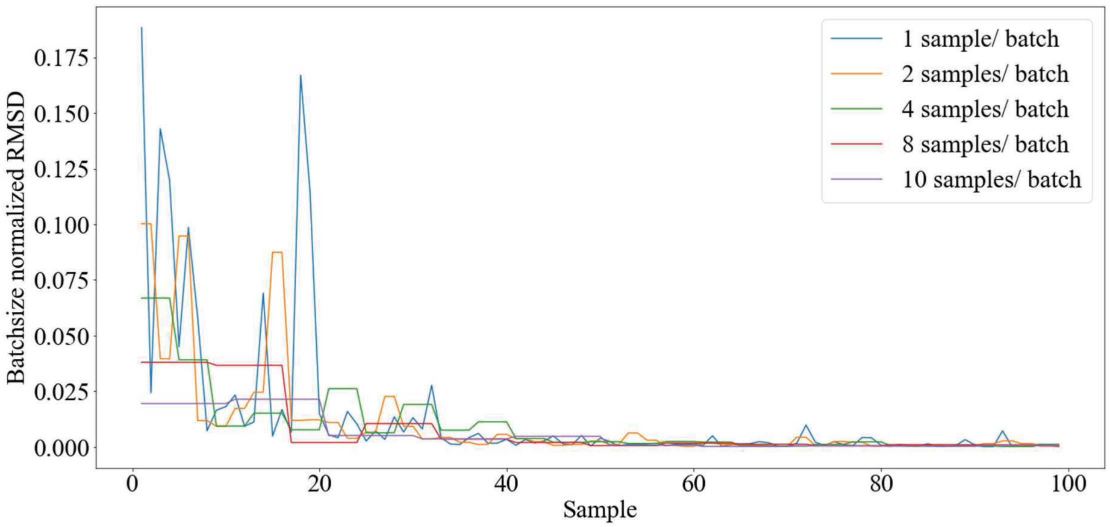


Figure 8. Plot of the RMSD values for multiple fixed (static) batch sizes normalized by batch size over the sample generation for the Hosaki test function.

Even in very late stages of the data generation process, small batch sizes still yield significantly higher relative RMSD values per sample than larger ones.

APPLYING THE PIPELINE TO GENERATE A CRASH SIMULATION TRAINING DATA SET

The FE Simulation Model – Pre-processing, Load case, Design space

The aim of project ATTENTION is to develop a framework and the constituting methods to predict the injury risk of VRUs in real-time using ML and simulation-based training data. This data is generated with a representative FE vehicle model [48] and the Total HUman Model for Safety (THUMS^{TM}) V4.02 AM50 Pedestrian [49] (see Figure 9). The overall project scope covers the extraction of vehicle-bound video data to determine parameters of accident scenarios, the generation of simulation-based training data as well as the training of advanced ML models aiming for the real-time prediction of situation-specific injury risks of VRUs. In future applications, this situation-specific injury risk prediction could support decision making in determining active risk minimization strategies.

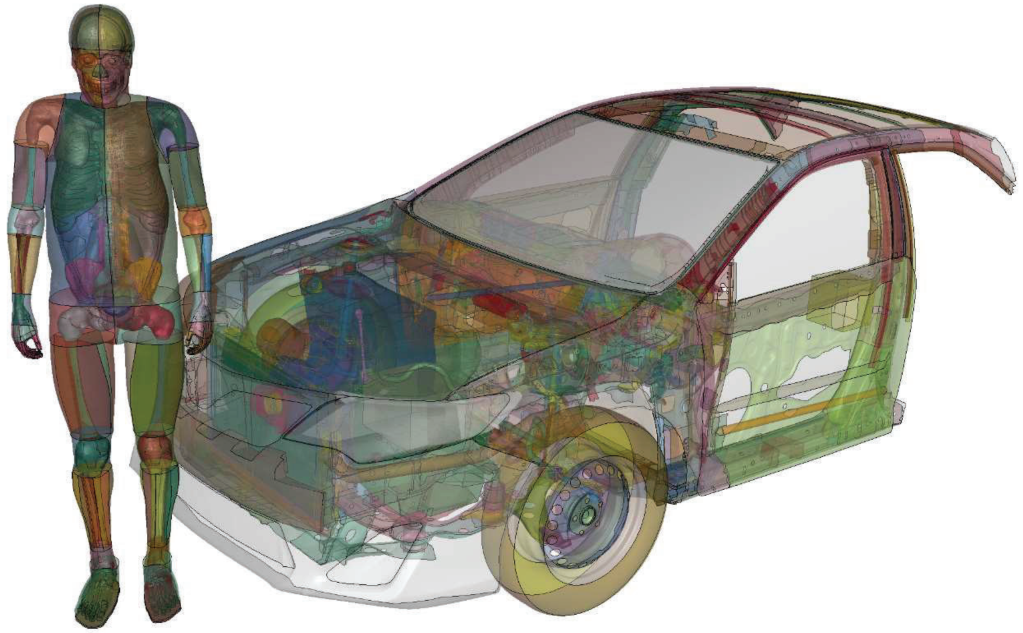


Figure 9. Full-scale simulation setup based on adapted Toyota Camry FE model and Total Human Model for Safety (THUMS^{TM}).

The first step of data generation is the preparation of the FE models. This includes a load-case-specific reduction of the vehicle model complexity in order to reduce computational costs and storage space while retaining important kinematic and dynamic properties as well as computational stability (see Figure 9). In the next step, parts with an extensive effect on the injury-relevant variables are analyzed in detail (e.g. windshield). Here, in particular, further effort is made to improve the predictive capability of the structural behavior of the windshield model, typically consisting of a PVB mid-layer and two outer glass layers (see Figure 10), making use of the latest results provided by the scientific community [50] [51] [52] [53] [54].

The PVB layer is modeled using solid elements and LS-Dyna *MAT_HYPERELASTIC_RUBBER (material data used from Osnes et al. [52], Jaware et al. [53] and Alter et al. [54]), the glass layers are modeled using shell elements and LS-Dyna * MAT_GLASS (material data used from Osnes et al. [52]). In similar fashion, the layers are connected using shared nodes for shell and solid elements. In order to bond the windshield to the body in white, an additional part, representing the bonding layer (material data used from [55]), is used at the windshield's edges in combination with LS-Dyna *CONTACT_TIED_NODES_TO_SURFACE similar to the connection used in the NHTSA Honda Accord Model [55].

Since this study focuses on the efficient generation and optimization of simulation-based training data using adaptive sampling methodology, an exemplary small-scale sub-model is extracted from the above-mentioned model setup (Toyota Camry, THUMS^{TM}) used in the ATTENTION project. This sub-model consists of the adapted windshield model impacted by a human head model isolated from the THUMS^{TM} model. The rear side of the bonding layer (initially attached to the body in white) is fixed in space. The THUMS^{TM} head is extracted retaining important instrumentation for injury measurement - such as accelerations, (angular) velocities and strains - which allows for the calculation of various injury criteria, such as HIC [56] [57], BrIC [58] and CSDM-calculation from the white and grey brain matter [59].

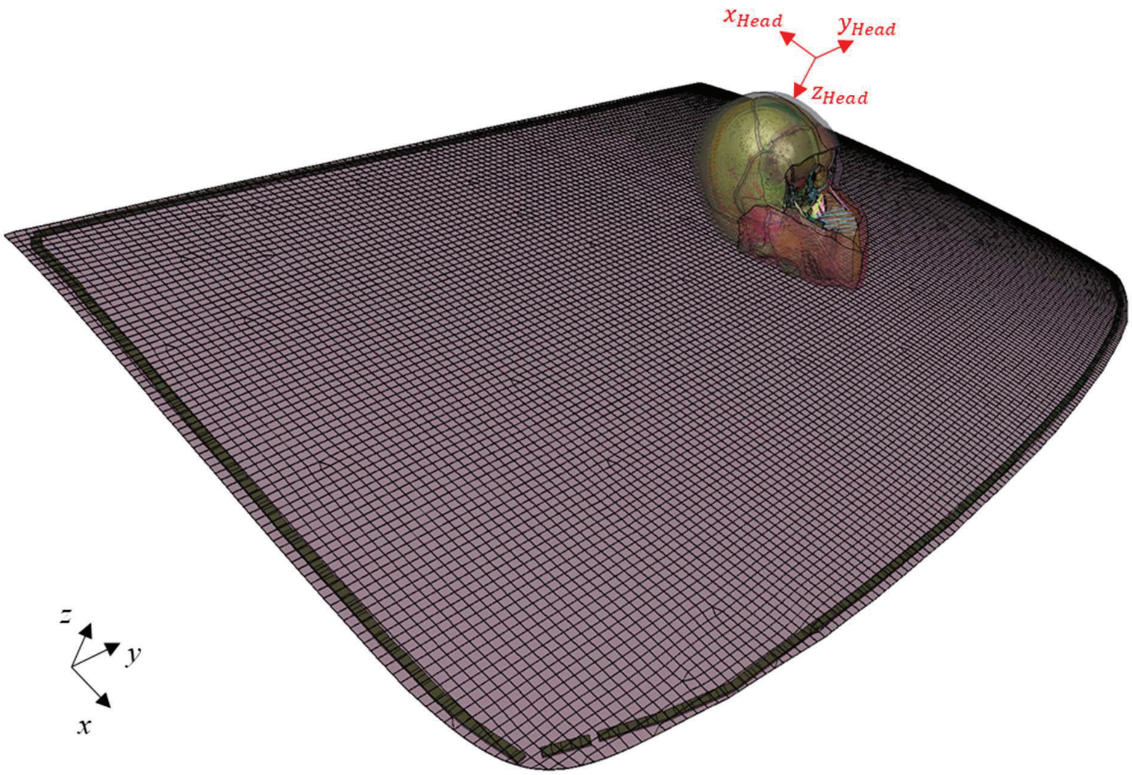


Figure 10. Sub-Model extracted from full-scale simulation model based on adapted windshield model and THUMS^{TM} Head. *(Nominal position: x-rotation = 0°, y-rotation = 270°, x-position = y-position = 0 mm)*

The considered parameter space is spanned by the parameters listed in Table 1. These parameters and their ranges are derived from a simulation study using the above-mentioned full-scale model considering information about relevant real-world accident scenarios provided in the GIDAS (German In-Depth Accident Study) database [60]. The termination time is defined dynamically based on loss of contact, calculation time using 32 CPUs is about 10 minutes. Defined output leads to a storage consumption of 3.5 GB per sample. The training data was generated using an adapted LS-DYNA MPP 9.3 Version on AMD EPYC 7763 CPU [39].

Table 1. Parameter space used during data generation considering the sub-model

Parameter	Minimum	Maximum
Head velocity	5 km/h	45 km/h
x-rotation (Head COS)	270°	360°
y-rotation (Head COS)	180°	270°
x-position (global COS)	-300 mm	370 mm
y-position (global COS)	-550 mm	0 mm

Adaptive Data Generation Cycles

As for the boundary conditions of the experiment, the evaluation points were defined by grid-sampling the design space with a total number 1000 equidistant points. The initial seed batch size was 5 and the maximum number of created samples was set to be 120. The GP regression model is adjusted to process 5 input parameters, but the actual model is the same as the one previously explained. For simplicity, the maximum resultant acceleration of a representative head node is chosen to be the output response for the GP model. As the initial goal is to compare the results with the results from applying the pipeline to the generic Hosaki function example, experiments with fixed batch sizes of 1, 2, 4, 8 and ten samples per batch were conducted. Given the high cost of calculating the ground truth for each evaluation point for RMSE and the limited availability of training samples in the initial iterations, an RF regressor with 100 trees is utilized along with 5-fold cross-validation. As the RF regressor is only used for model evaluation and not predictions, the white-box nature of the model could potentially be used for interpretability using input feature importance and proximity plots. The RF regressor used in this experiment can be replaced with any other ML model, including neural networks.

Figure 11 depicts the plot of RMSD and RMSE values over the generation of 120 samples with a batch size of 1. Similar to the example problem results in Figure 5, a global convergence is observed as the model approximates the maximum resultant acceleration response surface. The changes in the RMSD are seen to clearly resemble those in the RMSE, but the coherency observed in Figure 5 is more significant. In contrast to a fixed relative saturation criterion, considering a floating saturation criterion (using statistical metrics like mean and max over a particular number of iterations) may be of interest given the local spikes in RMSD.

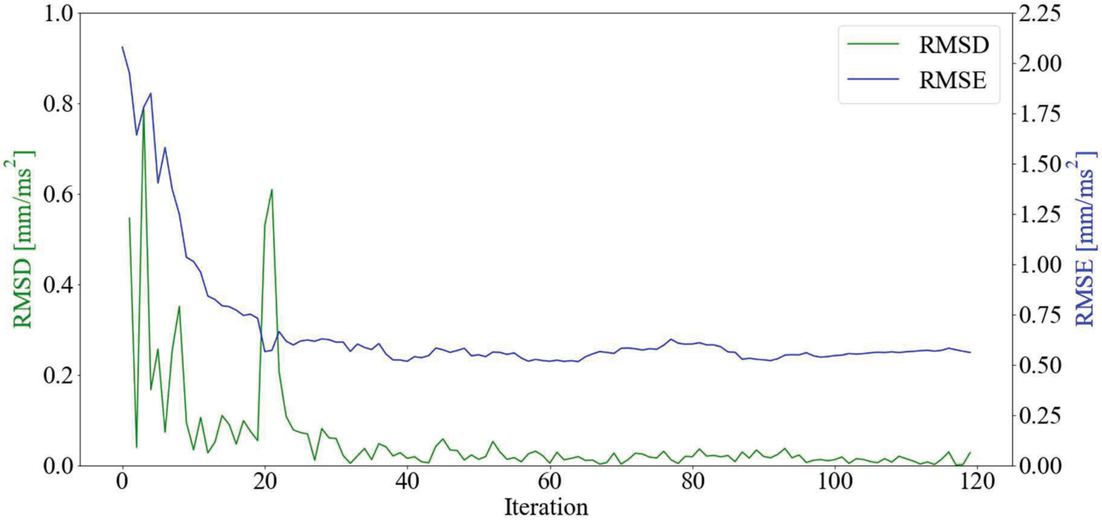


Figure 11. Plot of the RMSD and RMSE values over generated samples / iterations with batch size 1 for the sub-model (adapted windshield model and THUMS^{TM} Head).

Figure 12 depicts the plot of the RMSE over generated samples with multiple fixed batch sizes. Similar to the results in Figure 7, it can be observed that a smaller batch size results in a gradual and faster convergence towards lower error values. From the mutual convergence of the RMSE curves, it is also noted that, at a later stage, the batch size doesn't affect the overall model convergence, but a smaller batch sizes seem to have a better RMSE performance compared to bigger batch sizes.

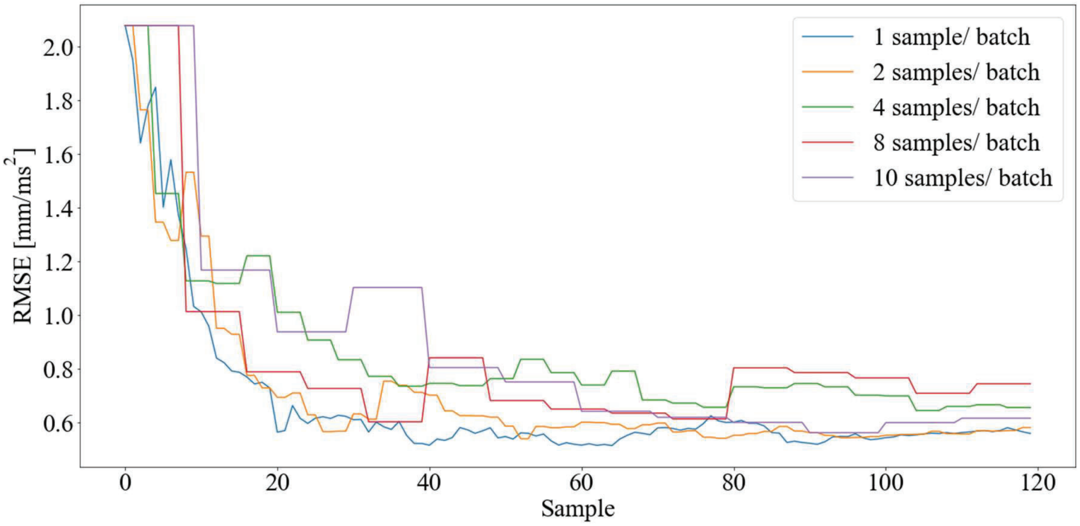


Figure 12. Plot of the RMSE over generated samples with multiple fixed batch sizes for the sub-model (adapted windshield model and THUMS^{TM} Head).

Figure 13 depicts the plot of the RMSD value normalized by batch size over the sample generation for multiple fixed batch sizes. Similar to the results shown in Figure 8, it can be observed that lower batch sizes yield higher values than larger batch sizes over the course of data generation process. Additionally, it is worth noting the similarities between batch sizes 1 and 2. This indicates the similar – and initially advantageous - behavior of smaller batch sizes with respect to the selection of "next best samples" based on the model variance and the existing body of information contained in the data set.

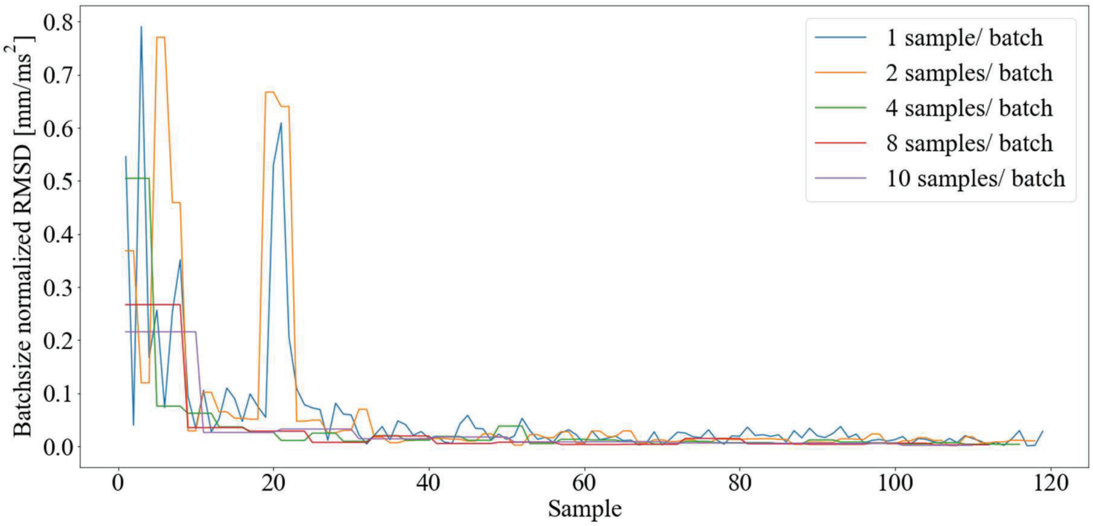


Figure 13. Plot of the RMSD values for multiple fixed batch sizes normalized by batch size over the sample generation for the sub-model (adapted windshield model and THUMS^{TM} Head).

In general, it can be subsumed, that the pipeline dynamics (relative rate of convergence, batch size effects, etc.) of the Hosaki example problem and the real engineering structural design application are very similar.

CONCLUSION

In order to successfully and sustainably apply ML methods for structural engineering tasks in vehicle or traffic safety, a robust approach towards efficiently generating training data with FE simulations is vital. A critical aspect to this success is the reusability of the generated data. Following general notions from the ML domain, one foundational hypothesis of this work is that data reusability in maximized with maximizing generalizability through the representational character of the data set from an expert perspective. As supported by the "no free lunch" theorem, no single ML algorithm universally performs the best for all the prediction tasks and the predictions quality strongly the quality of the data it is trained on [61]. This additionally motivates the key focus on the data itself and optimization of its representation of the system's complex behavior following a data-centric ML approach.

This study proposes a novel general pipeline architecture to generate data sets, which - within reasonable limits - represent all the relevant characteristic features of the system's behavior (crashworthiness characteristics) without tailoring it to a very specific ML application. After introducing the pipeline architecture with reference to existing solutions, it is applied to a generic mathematical example problem before applying it to a real vehicle safety application comprising FE simulation for data generation.

The result of this first implementation of the proposed architecture with a fixed batch size and no advanced termination criterion suggests that the pipeline can aid in generating a simulation data set that represents the relevant characteristic features of the system's behavior. The results reflect the expected behavior for the generic example and are confirmed in the real application scenario. This proves the general applicability of this novel pipeline architecture and supports the hypothesis regarding the significant potentials that lie in the multiple scaling routes and extensions of the same.

Technical conclusions include the confirmation of the potential of considering the RMSD as an indicator of convergence or the gradual saturation of information density in the data set. As indicated above, floating conditions might prove to work better than discrete threshold values as termination criteria. Furthermore, it is important to mention the strong effects of the meta model parameter settings. Considering the strong implications of changing the GP model kernel and the effects of using isotropic or anisotropic length scale parameters highlights the pitfalls of model dependent adaptive sampling processes as seen on conventional AL schemes. Using multiple model types simultaneously might thus be a good solution to avoid overlooking such issues.

As mentioned, this implementation should be considered the groundwork for future extensions and adaptions. One major task is to examine the overall performance of the proposed architecture by comparing the final prediction quality in the exploitation phase to models trained on uniformly (LHS) sampled training data sets and by evaluating transfer learning and data set extension/adaption approaches. This requires extensive study efforts, which will be prioritized in future works. Additional future extensions include employing dimensional reduction methods, such as learned manifold mode representations [62], in step 3 (data processing) to increase the information density of the data itself. Simultaneously employing multiple model types and their effects on "data set representativeness" in step 4 is also left to be studied in the following steps. Furthermore, the learnings regarding potentially beneficial strategies for the dynamic batch size adaption need to be implemented and studied. One of the main things to be implemented is the introduction of expert-driven metrics for evaluating the data set quality and – by that – steering the data generation process. As indicated, promising directions include the definition of dimension-specific sampling densities and of design space regions of increased relevance.

*The project ATTENTION is funded by the German Federal Ministry for Economic Affairs and Climate Action.

REFERENCES

[1] M. Haag, „Verkehr und Verkehrssicherheit im urbanen Raum,“ in *Freiburger Dialog*, Freiburg, 2022.

[2] W. Lerner, „The Future of Urban Mobility - Towards networked, multimodal cities of 2050,“ Arthur D. Little, 2011.

[3] F.-J. Van Audenhove, L. Dauby, O. Korniichuk und J. Pourbaix, „The Future of Urban Mobility 2.0 - Imperatives to shape extended mobility ecosystems of tomorrow,“ Arthur D. Little, 2014.

[4] D. Adminaité-Fodor und G. Jost, „How safe is walking and cycling in Europe?,“ European Transport Safety Council, Brussels, 2020.

[5] BaWü Statistisches Landesamt, „Mehr verunglückte Fahrradfahrende im ersten Halbjahr 2020.: Baden-Württemberg: Zahl der Getöteten dennoch rückläufig – Verunglückte mit Pedelecs legen um 47 % zu,“ Stuttgart, 2020.

[6] J. Moennich, T. Lich, A. Gerogi und N. Reiter, „Did a higher distribution of pedelecs results in more severe accidents in Germany,“ 2015.

[7] Fraunhofer EMI, „ATTENTION – Artificial intelligence for real-time injury prediction,“ 2021. [Online]. Available: https://www.emi.fraunhofer.de/en/news/news-press/attention----artificial-intelligence-for-real-time-injury-predic.html.

[8] M. Gonter, A. Leschke und U. Seiffert, „Fahrzeugsicherheit,“ in *Vieweg Handbuch Kraftfahrzeugtechnik, ATZ/MTZ-Fachbuch*, Springer, 2016, pp. 1105 - 1161.

[9] L. Greve, B. von de Weg und M. Andres, „Necking Prediction using Neural Networks,“ Baden Baden, 2019.

[10] C. Kohar, K. Inal, D. Kracker und P. Schwanitz, „Applications of artificial intelligence in automotive engineering for crashworthiness design,“ in *VDI Conference - Automotive CAE*, Baden Baden, 2019.

[11] B. Van de Weg, „Surrogate modeling for finite element simulations,“ in *SIMVEC*, Baden Baden, 2022.

[12] Landing AI, „What is data-centric AI?,“ 2022. [Online]. Available: https://landing.ai/data-centric-ai/.

[13] A. Ng, „How AI is changing the future of business,“ Berlin, 2022.

[14] S. Vasu, N. Talabot, A. Lukoianov, P. Baque, J. Donier und P. Fua, „HybridSDF: Combining Free Form Shapes and Geometric Primitives for effectiveShape Manipulation,“ arxiv, 2021.

[15] L. Greve und B. Van de Weg, „Surrogate modeling of parametrized finite element simulations with varying mesh topology using recurrent neural networks,“ *Array,* Nr. 14, 2022.

[16] L. Greve, „KI-basierte Echtzeitmodelle für die schnelle Materialparameterkalibrierung,“ Freiburg, 2021.

[17] C. Kohar, L. Greve, T. Eller, D. Conolly und K. Inal, „A machine learning framework for accelerating the design process using CAE simulations: An application to finite element analysis in structural crashworthiness,“ *Computer Methods in Applied Mechanics and Engineering,* Nr. 385, 2021.

[18] L. Chec, „How Machine Learning and AI accelerates automotive design processes. Feedback on 3 different application: battery design, structural optimization and system design,“ Hanau, 2022.

[19] K. Kayvantash, „A study on the effect of sampling on the quality of AI/ML/ROM models,“ Hanau, 2022.

[20] B. Settles, „Active Learning Literature Survey - Computer Sciences Technical Report 1648,“ University of Wisconsin–Madison, Madison, 2009.

[21] Y. Yoo, C.-K. Park und J. Lee, „Deep learning-based efcient metamodeling via domain knowledge-integrated designable data augmentation with transfer learning: application to vehicle crash safety,“ *Structural and Multidisciplinary Optimization,* Bd. 65, 2022.

[22] J. Brownlee, „MachineLearningMastery - A Gentle Introduction to Transfer Learning for Deep Learning,“ 16 September 2019. [Online]. Available: https://machinelearningmastery.com/transfer-learning-for-deep-learning/.

[23] M. McKay, R. Beckman und W. Conover, „A Comparison of Three Methods for Selecting Vales of Input Variables in the Analysis of Output From a Computer Code,“ Bd. Technometrics , Nr. 21, 1979.

[24] F. Duddeck, „Robust Design for Car Body Development - Design of Experiments DoE,“ Freiburg, 2019.

[25] I. Goodfellow, Y. Bengio und A. Courville, „Machine Learning Basics,“ in *Deep Learning*, Cambridge, MIT Press, 2016, pp. 98 - 155.

[26] I. Goodfellow, Y. Bengio und A. Courville, „Sequence Modeling: Recurrent and Recursive Nets,“ in *Deep Learning*, Cambridge, MIT Press, 2016, pp. 373 - 416.

[27] D. Rumelhart, G. Hinton und R. Williams, „Learning representations by back-propagating errors,“ *Nature,* pp. 533 - 536, 1986.

[28] M. Turek, „Explainable Artificial Intelligence (XAI),“ [Online]. Available: https://www.darpa.mil/program/explainable-artificial-intelligence.

[29] R. Roscher, B. Bohn, M. Duarte und J. Garcke, „Explainable Machine Learning for Scientific Insights and Discoveries,“ *IEEE Access,* pp. 42200 - 42216, 2019.

[30] A. Géron, Hands-On-Machine-Learning-with-Scikit-Learn-Keras-and-Tensorflow, Sebastopol. CA: O'Reilly Media Inc., 2019.

[31] B. Settles, M. Craven und S. Ray, „Multiple-Instance Active Learning,“ in *Advances in Neural Information Processing Systems (NIPS)*, Boston, 2008.

[32] A. Freytag, E. Rodner und J. Denzler, „Selecting Influential Examples: Active Learning with Expected Model Output Changes,“ in *Proceedings of the 13th European Conference on Computer Vision, ECCV 2014*, Zürich, 2014.

[33] E. Pasolli und F. Melgani, „Gaussian process regression within an active learning scheme,“ in *2011 IEEE International Geoscience and Remote Sensing Symposium*, Vancouver, 2011.

[34] D. Cohn, Z. Ghahramani und M. Jordan, „Active Learning with Statistical Models,“ *Journal of Artifical Intellgence Research,* Bd. 4, pp. 129 - 145, 1996.

[35] T. Lookman, P. Balachandran, D. Xue und R. Yuan, „Active learning in materials science with emphasis on adaptive sampling using uncertainties for targeted design,“ *npj Computational Materials,* Bd. 5, Nr. 21, 2019.

[36] V. Aute, K. Saleh, O. Abdelaziz und S. Azarm, „Cross-validation based single response adaptive design of experiments for Kriging metamodeling of deterministic computer simulations,“ *Struct Multidisc Optim,* Bd. 48, pp. 581-605, 2013.

[37] B. Maschler, H. Vietz, H. Tercan, C. Bitter, T. Meisen und M. Weyrich, „Insights and Example Use Case on Industrial Transfer Learning,“ in *55th CIRP Conference on Manufacturing Systems*, Lugano, 2022.

[38] M. Liefvendahl und R. Stocki, „A study on algorithms for optimization of Latin hypercubes,“ *Journal of Statistical Planning and Inference,* Bd. 136, Nr. 9, pp. 3231 - 3247, 2006.

[39] Livermore Software Technology Corporation, „LS-DYNA R10.1,“ Livermore, 2022.

[40] Lasso GmbH, „Lasso Python Library GmbH,“ Lasso, [Online]. Available: https://lasso-gmbh.github.io/lasso-python/build/html/. [Zugriff am 22 December 2022].

[41] The pandas development team, „pandas-dev/pandas: Pandas,“ February 2020. [Online]. Available: https://doi.org/10.5281/zenodo.3509134. [Zugriff am 22 December 2022].

[42] C. E. Rasmussen, „Gaussian Processes in Machine Learning,“ in *Advanced Lectures on Machine Learning, Lecture Notes in Computer Science*, Berlin, Heidelberg, Springer, 2004, p. 63–71.

[43] „GPy: A Gaussian Process Framework in Python,“ 2012. [Online]. Available: http://github.com/SheffieldML/GPy.

[44] M. Kuhn und K. Johnson, „Measuring Performance in Regression,“ in *Applied predictive modeling*, Springer, 2018, pp. 95-100.

[45] M. Kuhn und K. Johnson, in *Applied predictive modeling*, Springer, 2018, pp. 69-73.

[46] R. Jin, W. Chen und A. Sudjianto, „On sequential sampling for global metamodeling in engineering design,“ *International design engineering technical Conferences and Computers and Information in engineering conference,* pp. 539-548, 2002.

[47] G. A. Bekey und M. T. Ung., „A Comparative Evaluation of Two Global Search Algorithms,“ *IEEE Trans. Syst. Man Cybern.,* pp. 112-116., 1974.

[48] Center for Collision Safety and Analysis at the George Mason University (GMU), Federal Highway Administration (FHWA), „2012 Toyota Camry Detailed Finite Element Model,“ https://www.ccsa.gmu.edu/models/2012-toyota-camry, 2012.

[49] K. Shigeta, Y. Kitagawa und T. Yasuki, „Development of Next Generation Human FE Model capable of Organ Injury Prediction,“ in *Proceedings of the 21st Annual Enhanced Safety of Vehicles*, Stuttgart, 2009.

[50] N. Kulkarni, S. Deshpande und R. Mahajan, „Development of Pedestrian Headform Finite Element Model using LS-DYNA and its Validation as per AIS 100/GTR9,“ in *12th European LS-Dyna Conference*, Koblenz, 2019.

[51] J. Prasongngen, I. Putra, S. Koetniyom und J. Carmai, „Improvements of Windshield Laminated Glass Model for Finite Element Simulation of Head-to-Windhsield Impacts,“ in *IOP Conference*, Phuket, 2013.

[52] K. Osnes, S. Kreissl, J. D'Haen und T. Borvik, „Modelling of Fracture Initiation and Post-Fracture Behaviour of Head Impact on Car Windshields,“ in *13th European LS-Dyna Conference*, Ulm, 2021.

[53] A. Jaware, S. Chandratre, M. Perez und J. Narule, „An Advanced Methodology for Windscreen Modeling in LS Dyna,“ *International Journal of Mechanical Engineering and Technology (IJMET),* Bd. 10, 2019.

[54] C. Alter, S. Kolling und J. Schneider, „A new failure criterion for laminated safety glass,“ in *11th European LS-Dyna Conference*, Salzburg, 2018.

[55] National Highway Traffic Safety Administration (NHTSA), „Honda Accord FE-Model,“ NHTSA, https://www.nhtsa.gov/crash-simulation-vehicle-models.

[56] M. Kleinberger, E. Sun und R. Eppinger, „Development of improved injury criteria for the assessment of advanced automotive restraint systems,“ in *NHTSA Docket 4405.9*, 1998.

[57] E. Herzt, „A note on the head injury criterion (HIC) as a predictor of the risk of skull fracture,“ in *Proceedings: Association for the Advancement of automotive medicine annual conference*, Des Plaines, 1993.

[58] E. Takhounts, M. Craig, K. Moorhouse, J. McFadden und V. Hasija, „Development of brain injury criteria (BrIC),“ in *Stapp car crash journal 57*, Orlando, 2013.

[59] E. Takhounts, R. Eppinger, Q. Campbell, R. Tannoues, E. Power und L. Shook, „On the Development of the SIMon Finite Element Head Model," in *Stapp Car Crash Journal Vol. 47*, San Diego, 2003.

[60] Verkehrsunfallforschung an der TU Dresden GmbH, „GIDAS - German In-Depth Accident Study," 2022. [Online]. Available: http://gidas.org/.

[61] D. H. Wolpert und W. G. Macready, „No free lunch theorems for optimization," *IEEE Transactions on Evolutionary Computation ,* Bd. 1, Nr. 1, pp. 67 - 82, 1997.

[62] J. Garcke und R. Iza-Teran, „Machine Learning Approaches for Data from Car Crashes and Numerical Crash Simulation," in *NAFEMS World Congress*, Stockholm, 2017.

PREDICTION OF ALL RIB DEFLECTIONS OF THOR-ATD BY MEANS OF DEEP NEURAL NETWORK MODEL

Takayuki Kawabuchi
Yasuhiro Dokko
Honda R&D Co., Ltd.
Japan

Hidenori Mikami
Honda Motor Co., Ltd
Japan

Kota Katsushima
IDAJ Co., LTD.
Japan

Yosuke Nagai
Photron Limited
Japan

Paper Number 23-0219

ABSTRACT

The fatality rate of thoracic injury for elderly occupants in vehicle accidents is significantly high. Its major cause is the rise of internal organ injury rates due to an increase in the number of fractured ribs (NFR). Therefore, NFR reduction is crucial to enhance elderly occupant protection and is one of the key issues for achieving zero fatalities. In order to improve NFR prediction accuracy, the previous study proposed the criterion using the weighted averaged displacement of all ribs (WADAR), which indicated a higher correlation coefficient with NFR than that of the criterion, Rmax, using four Infra-Red Telescoping Rod for the Assessment of Chest Compression (IR-TRACC) installed on the thorax of the Test device for Human Occupant Restraint Anthropometric Test Dummy (THOR-ATD). While WADAR requires all rib deflections, it is difficult to install IR-TRACCs on all ribs inside the limited space in the thorax of THOR-ATD. The objective of this research is to predict the deflections of all ribs by means of a neural network model using time-histories of rib deflections from four IR-TRACCs and the crash velocity without any installation of additional measurement devices.

The architecture of the neural network model is based on U-Net, which is one of the convolutional neural network models. The model was trained by time-historical X, Y and Z displacements of 14 ribs and the crash velocity derived from the 56 FEM simulation data, which represented frontal and oblique sled experiments with THOR-ATD. The model learned the physical relationships among the ribs with and without IR-TRACCs. The predicted rib deflections were validated by the THOR-ATD experiment, where the displacements of the 2nd to 6th ribs on the left side were measured three-dimensionally by the set of two cameras installed on the upper and lower thoracic spines.

The predicted deflections during 0 to 150 ms were processed into a resultant deflection and compared to the actual deflection through the 2nd to 6th ribs on the left side. The maximum differences in the peak deflection were 2.3 mm, respectively. Furthermore, the root mean square error (RMSE) was calculated at each rib for prediction accuracy evaluation, which resulted in minimum and maximum RMSE of 0.6 mm and 2.7 mm, respectively.

Although the number of training datasets was small, the neural network model trained by FEM simulation data could predict all the rib deflections with small error without physical measurement devices.

INTRODUCTION

Fatal thoracic injuries in frontal crashes appeared with frequency equal to, or following, head fatal injuries [1]. Kent et al. reported that the percentages of drivers who died with injuries related to rib fractures increased with aging and suggested that rib fracture was associated with the significantly increasing fatality rate of thoracic injuries, especially in elderly occupants [2]. It is estimated that the population of adults over 65 years old will increase up to 83.7 million by the year 2050 in the United States [3] and it will result in an increasing number of drivers sustaining severe injury to the thorax in traffic accidents.

Since the number of fractured ribs (NFR) is correlated with a rise in a fatality rates of the elderly population [4][5][6], the criteria predicting NFR with high accuracy are necessary for the development of an occupant protection device.

Kent et al. suggested that the risk of rib fractures increased with the level of thoracic compression and the thoracic injury risk was often described by the antero-posterior deflection of the thorax [7]. The thoracic deflection is measured by sensors installed in the thorax of the Test device for Human Occupant Restraint Anthropometric Test Dummy (THOR-ATD). THOR-ATD has four Infra-Red Telescoping Rods for the Assessment of Chest Compression (IR-TRACC) and they are often used to estimate the injury level by criteria such as Rmax [8], which uses maximum resultant deflection, and PCscore [9] which is calculated by the formula based on the primary component analysis by means of four IR-TRACCs values.

Kawabuchi et al. reported that NFR increased without deformation at the ribs with IR-TRACCs when a region remote from those ribs, such as the clavicle or upper part of the rib cage, were impacted. Under such conditions, the criteria using weighted averaged displacement of all ribs (WADAR) correlated better with NFR than that of the other criteria such as Rmax [10]. Whereas WADAR requires all rib deflections, it is difficult to install IR-TRACCs on all ribs inside the limited space in the thorax of THOR-ATD. Hence, this study suggests a predictive solution instead of physical measurement devices.

Recent studies have investigated the application of physical simulation results to deep learning. Guo et al. developed the Deep Neural Network (DNN) model, which predicted a steady flow field by means of the latent fluid characteristics learned from computational fluid dynamics simulation results [11]. Ito et al. constructed a model that predicted pedestrian injury values by means of pedestrian crash simulation results [12]. As indicated in the previous studies, a DNN could learn a physical relationship between multiple outputs such as trajectories from the simulation results. For example, when the simulation well reconstructs the actual physical environment, the model trained by the simulation data predicts movement of one location in the actual experiment results from another separated location based on the latent physical relationship.

In order to construct such a DNN model, the following are prerequisites. First, the multiple outputs of the simulation results mutually interact based on common physical relationships. Second, the simulation well models the actual physical environment. Since 14 thoracic ribs of THOR-ATD are connected by a rubber bib, the all ribs move together with the four ribs equipped with IR-TRACC based on the physical characteristics. That is, all rib deflections may be predictable from the time-historical deflection data measured by four IR-TRACCs. Also, the specific mechanical properties applied in the finite elemental (FE) model of THOR-ATD were validated by the calibration test results. As above, since the simulation data used in this study conformed to the two prerequisites, it may enable the DNN model to learn the latent physical relationships between the movement of the ribs with and without IR-TRACCs from the FE simulation results. Moreover, the model may predict all rib deflections in physical THOR-ATD from waveforms measured by IR-TRACCs.

The objective of this study is to develop a DNN model learning latent mechanical properties from THOR-ATD simulation data and to predict the time-historical deflections of all ribs by means of four IR-TRACC deflection data.

METHOD

Structure of THOR-ATD

Figure 1 shows the structure of the thoracic part of THOR-ATD. The thoracic part represents the chest of a human body and consists of a sternum and 14 ribs, which are fewer than the 24 ribs of an actual rib cage. The ribs and the sternum are connected by the costal cartilage, which is represented by the part called the bib in THOR-ATD. The tip of each rib is bolted onto the bib, and they are connected with the sternum plate. The sternum plate is divided into upper and lower parts, which are connected with the clavicle and 2nd to 7th ribs through the bib, respectively.

The IR-TRACCs consist of expandable tubes installed on the spine box and they are bolted on the respective tips of the 3rd and 6th ribs. An infrared ray is emitted in the tube and measures the change in rib deflections. The base of IR-TRACC on the spine box has Y and Z rotation axes with angle meters which make the tube move three-dimensionally. The spine box consists of upper and lower parts which are connected by the rubber block reproducing the bending movement of an actual spine. The 1st to 4th and 5th to 7th ribs are deformable steel plate with rubber damping materials and they are bolted onto the upper and lower spine boxes, respectively.

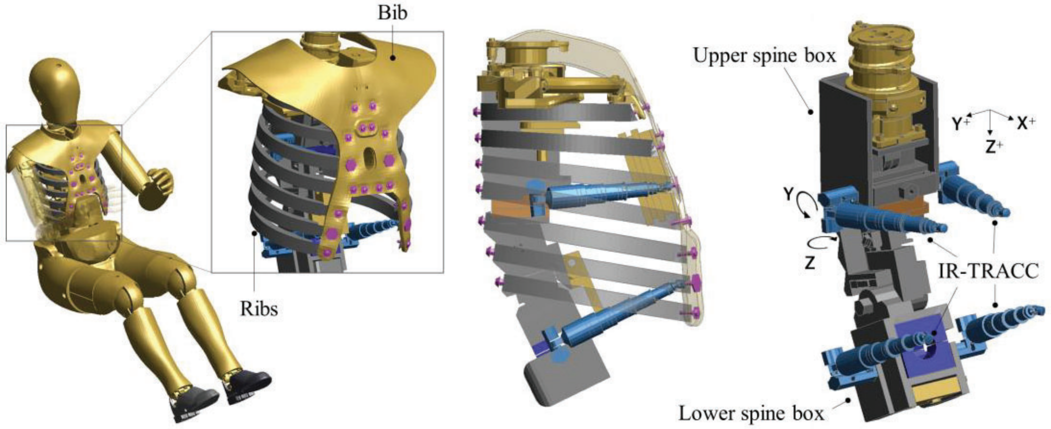


Figure 1. Illustration of the thoracic structure of THOR-ATD

Boundary Condition of FE Simulation for Training Data

The results of FE simulation (LS-DYNA R9.1.2) were utilized for the training data in this study. Figure 2 shows the boundary condition of the simulation, which reconstructs the frontal crash sled experiment conducted in European research project (SENIORS) [13]. The test rig had a steel pan for a seat, a seat belt, foot rests and a generic airbag by means of THOR-ATD FE model version 1.3.2 [14]. The table A1 shows the load cases with the test parameters: impact speed, impact angle and with/without airbags. The 150 ms time-historical X, Y and Z deflections of rib tips and the crash velocity were extracted from the 56 simulation results as training data.

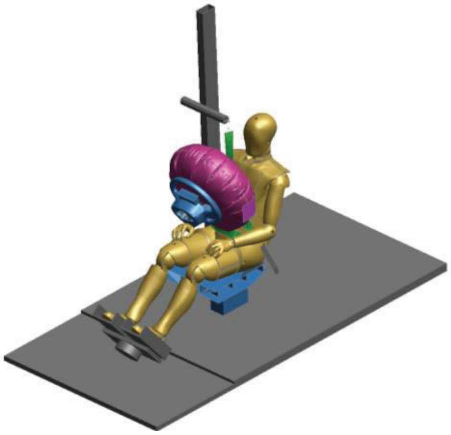


Figure 2. Illustration of the test rig

Deep Neural Network Model

The DNN model was constructed using the sled velocity and the X, Y and Z time-historical rib deflections of four ribs (13 waveforms) that can be measured experimentally by IR-TRACCs, which outputted the X, Y and Z time-historical rib deflections of 14 ribs (42 waveforms), including the four rib deflections that can be measured. The sled velocity was considered to contain physical characteristics of crash modes and indirectly associated with thoracic deformation patterns. Figure 3 shows the processing concept of the model.

U-Net [15] was used for the architecture of the DNN in this study, which was suitable for segmentation and style transformation and is often used in image processing. Figure 4 shows the conceptual diagram of the U-Net structure used in this study. U-Net that is based on Autoencoder with skip-connections enables reconstruction of images or waveforms more accurately than those by Autoencoder. Skip-connections transfer local information conventionally lost in the encoding process to the decoding process, improving prediction accuracy. In this study, U-net was assumed to be suitable because the required task was to transform the style from a known to an unknown waveform, rather than to predict an unknown waveform.

The training and test of the model were conducted by means of 56 FE simulation results described above and three experimental data, respectively. The three test data were boundary conditions similar to the training data as shown in the Design of Experiments (DOE) in Figure B1. The first and second tests, Test 1 and Test 2, respectively, were extracted from the published sled experiments. Test 1 was engaged in the SENIORS project [13], which utilized the test rig consistent with FE simulation. Test 2 was engaged in the University of Virginia [16], which was conducted using Taurus with rear seat equipped with a seatbelt without a load limiter. The third test (Test 3) was conducted at 40 km/h with the test rig processed from a mass-produced small size sedan model. As described in a later section, some of the deflections of ribs without IR-TRACCs were measured in the experiment by means of the optical method.

The loss function for the training of the neural network model was the combination of Mean Squared Error (MSE) and L1 regularization in order to prevent overfitting because of the small size of the training datasets in this study. Moreover, Root MSE (RMSE) and differences of peak resultant deflection, utilized as chest injury criteria such as Rmax and PCscore, were also calculated for the discussion of prediction accuracy, because those dimensioned criteria were more understandable about the amount of the error than that indicated by dimensionless MSE. Smaller values for these error indicators indicated smaller errors. Test 1 and Test 2 were utilized to verify that the DNN model could reconstruct the waveform from the IR-TRACC data as inputs. The loss functions of these two tests were calculated only on the measurable ribs by IR-TRACCs. On the other hand, Test 3 was utilized to verify that the deflection of ribs without IR-TRACCs could be predicted from the time-historical rib deflections of measurable ribs with IR-TRACCs and sled velocities.

Waveform data was preprocessed by standardization, i.e., setting the mean to 0 and the variance to 1. The calculation of the mean and variance for standardization was performed on simulation data alone.

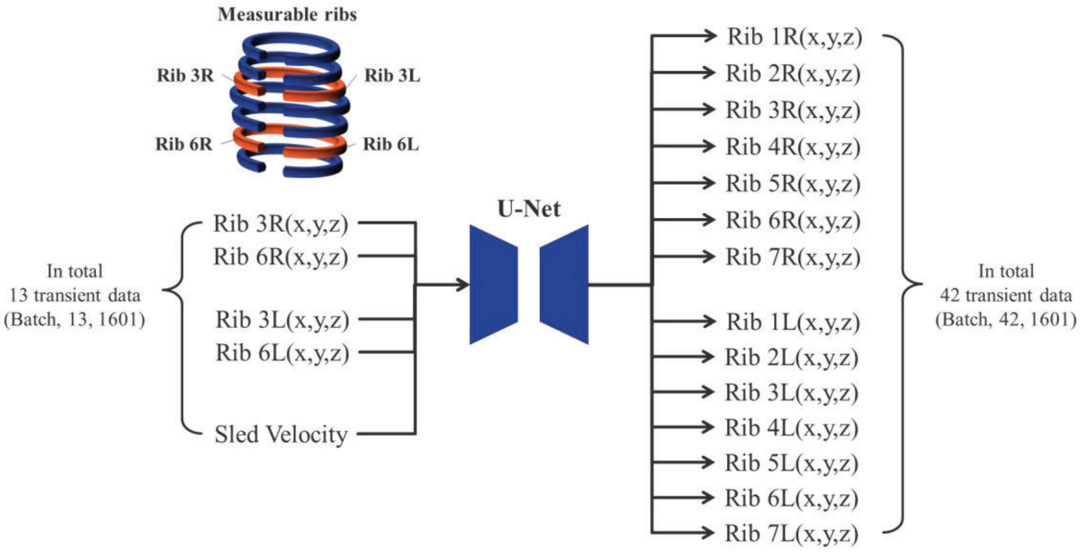
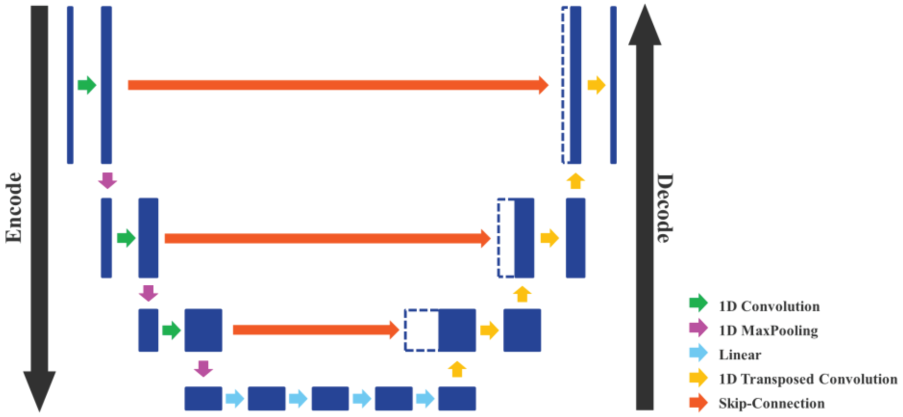


Figure 3. Concept of the model using U-Net



	Layer type	Activation function	BatchNormalization	Dropout ratio	(Channel size, Kernel size, Stride, Padding)	Output size
Encoder	1D Convolution	ReLU	1D BatchNorm	-	(32, 120, 3, 100)	(Batch, 32, 561)
	1D MaxPooling	-	-	-	-	(Batch, 32, 280)※1
	1D Convolution	ReLU	1D BatchNorm	-	(64, 25, 2, 0)	(Batch, 64, 128)
	1D MaxPooling	-	-	-	-	(Batch, 64, 64) ※2
	1D Convolution	ReLU	1D BatchNorm	-	(128, 5, 2, 0)	(Batch, 128, 30)
	1D MaxPooling	-	-	-	-	(Batch, 128, 15) ※3
	Linear	ReLU	1D BatchNorm	-	-	(Batch, 512)
	Linear	Tanh	-	-	-	(Batch, 30)
Decoder	Linear	ReLU	1D BatchNorm	-	-	(Batch, 512)
	Linear	ReLU	1D BatchNorm	-	-	(Batch, 1920)
	Concatenation※3	-	-	-	-	(Batch, 256, 15)
	1D Transposed Convolution	ReLU	1D BatchNorm	-	(128, 4, 2, 1)	(Batch, 128, 30)
	1D Transposed Convolution	ReLU	1D BatchNorm	-	(128, 6, 2, 0)	(Batch, 64, 64)
	Concatenation※2	-	-	-	-	(Batch, 128, 64)
	1D Transposed Convolution	ReLU	1D BatchNorm	0.5	(128, 4, 2, 1)	(Batch, 64, 128)
	1D Transposed Convolution	ReLU	1D BatchNorm	0.5	(128, 26, 2, 0)	(Batch, 64, 280)
	Concatenation※1	-	-	-	-	(Batch, 96, 280)
	1D Transposed Convolution	ReLU	1D BatchNorm	-	(128, 4, 2, 1)	(Batch, 64, 560)
	1D Transposed Convolution	ReLU	1D BatchNorm	-	(128, 124, 3, 0)	(Batch, 42, 1801)
	Cropping	-	-	-	-	(Batch, 42, 1601)

※1, 2, 3 Skip connection

Figure 4. Network architecture diagram and topology of U-Net

Stereoscopic Vision Measurement of the Ribs Without IR-TRACCs

The time-historical deflections of ribs without IR-TRACCs were measured by the stereoscopic vision measurement method in order to test the predicted results. Figure 5 shows the illustration of two sets of the stereo cameras installed on the spine box, which recorded stereoscopic vision for the three-dimensional measurement process. The set of twin cameras was assembled with two high speed cameras (FASTCAM MH6 ST-Cam, Photron, Japan) and its recording frequency was 1000 Hz. The target markers were stuck onto each rib tip. The upper and lower cameras measured the deflections of the 2nd to 3rd ribs and 3rd to 6th ribs, respectively. It was required for the recorded objects and cameras to belong to a common coordinate system. However, the 4th rib on the upper spine box was recorded by lower cameras which were installed on the lower spine box because the upper IR-TRACC was prevented from recording the 4th ribs by the upper cameras. Therefore, the optical measured deflection of the 4th rib was estimated by Equation 1.

$$Estimated\ 4^{th}\ rib\ deflection = \ 4^{th}\ rib\ deflection\ \times \frac{Upper\ 3^{rd}\ rib\ deflection}{Lower\ 3^{rd}\ rib\ deflection} \qquad Equation\ (1)$$

The stereoscopic vision measurement was engaged on the left-side ribs alone due to the mounting space limitation of the equipment. The THOR-ATD with the two sets of stereo cameras was installed on the sled test rig representing a small sized sedan and impacted at 40 km/h for data aggregation. The validation accuracy was evaluated by RMSE and differences of peak resultant deflection.

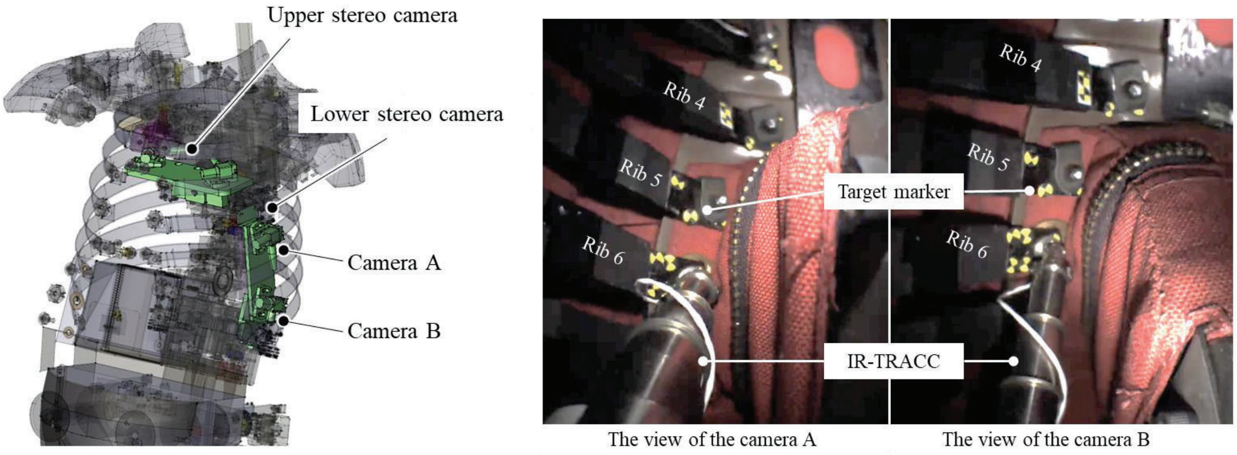


Figure 5. Installation of stereo cameras on the spine box

RESULTS

Learning of the Model

Figure 6 shows the MSE losses of the training and the test and those losses were less than 0.0067 and 0.019, respectively. Figures 7 and 8 show comparisons of the exact and predicted data of Test 1 and Test 2, respectively, and these indicated that the predicted rib deflections overall reconstructed the trend and peaks of exact waveforms. The average RMSE of each test result was less than 1.3 mm. Table 1 shows the RMSE and the differences of the peak resultant deflection of Test 1 and Test 2.

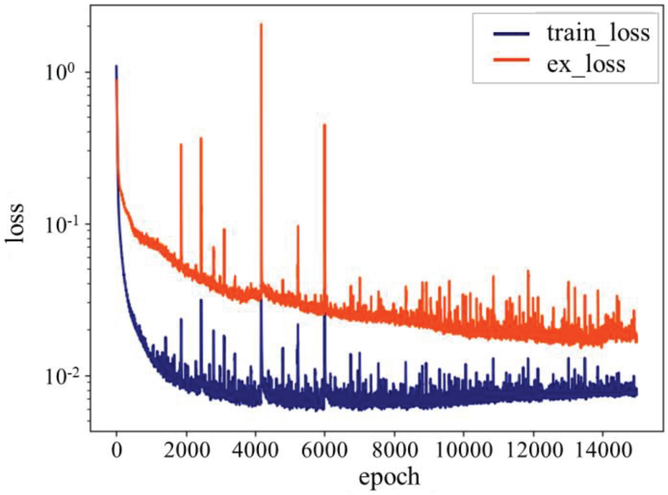


Figure 6. History of MSE loss function

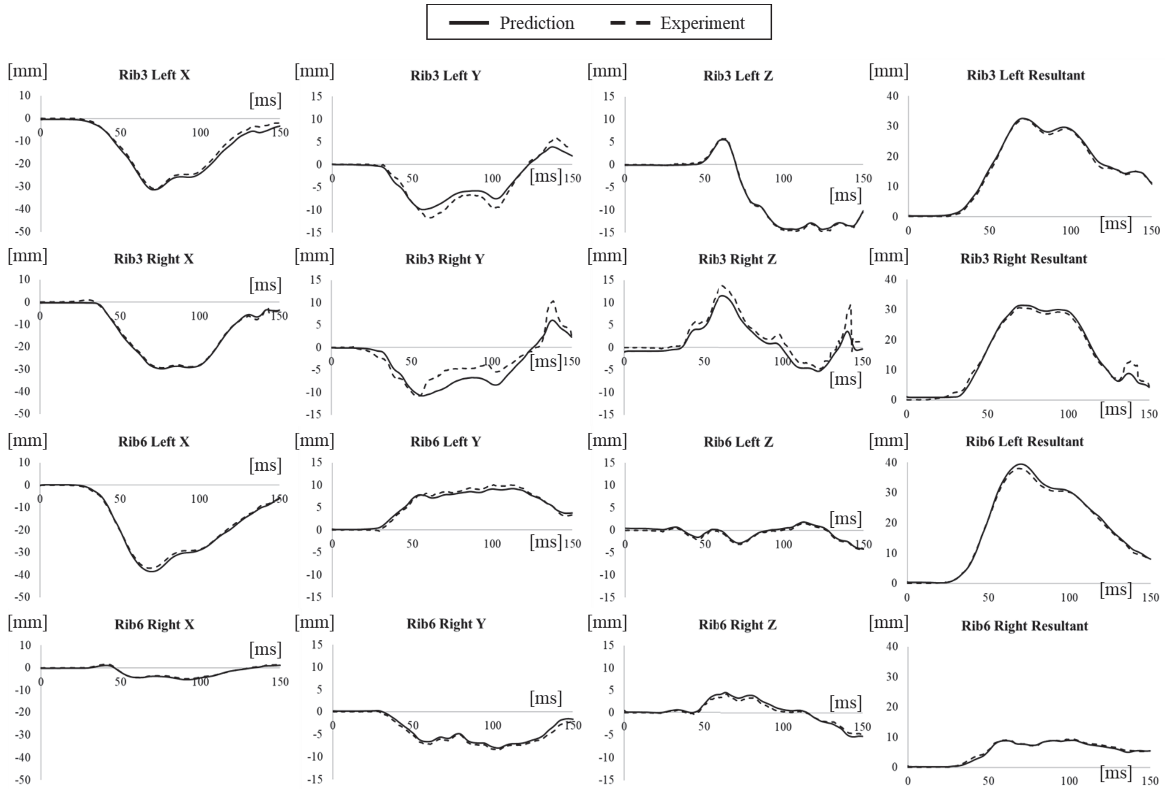


Figure 7. Comparison between predicted and exact rib deflections in Test 1

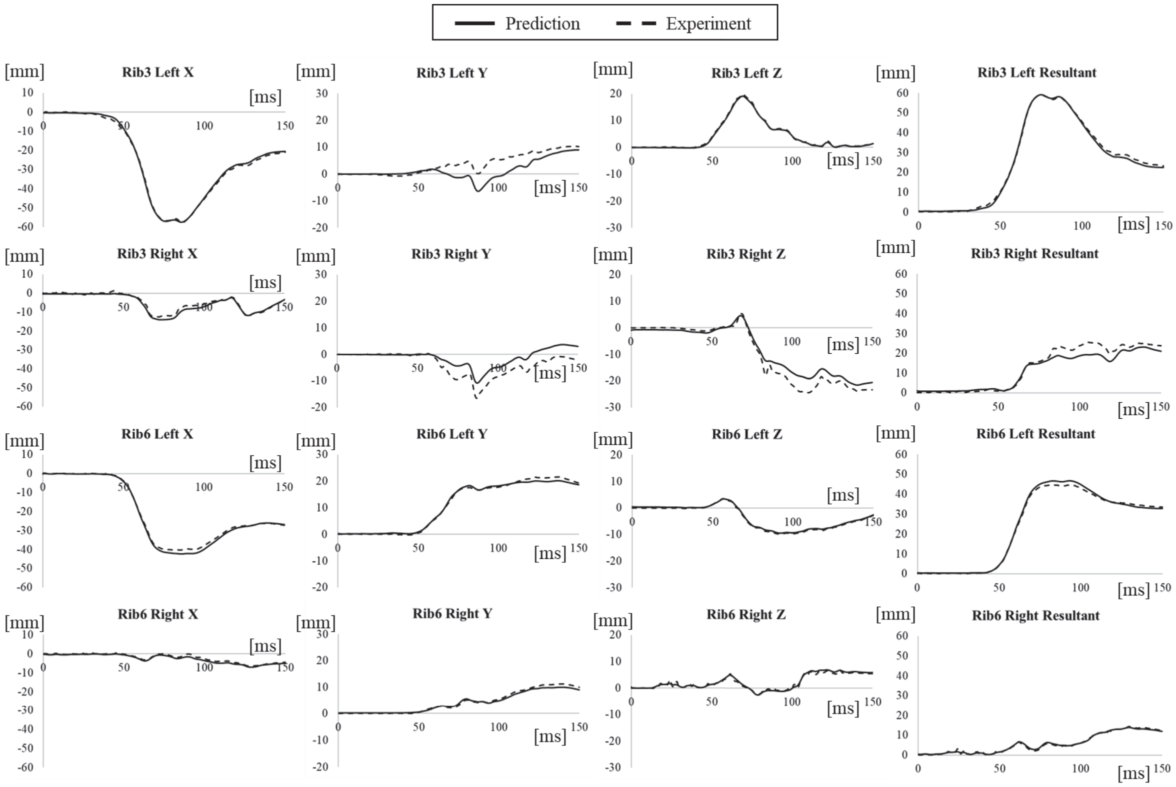


Figure 8. Comparison between predicted and exact rib deflections in Test 2

Table 1.
The RMSE and differences of the peak resultant deflection of Test 1 and Test 2

			X [mm]	Y [mm]	Z [mm]	Resultant [mm]	Differences of Peak Res. between Pred. and Exp. [mm]
Test 1	**Left**	**Rib 3**	1.3	1.1	0.2	1.0	-0.1
		Rib 6	0.8	0.5	0.3	0.7	-1.3
	Right	**Rib 3**	0.6	2.0	1.4	1.3	-0.7
		Rib 6	0.3	0.5	0.4	0.3	0.3
Test 2	**Left**	**Rib 3**	0.7	3.5	0.3	1.0	0.2
		Rib 6	1.0	0.7	0.3	1.0	2.3
	Right	**Rib 3**	0.9	3.2	2.6	2.7	-1.8
		Rib 6	0.6	0.6	0.4	0.4	0.6

The Results of Stereoscopic Vision Measurement

Figure 9 shows a comparison of the deflections measured by IR-TRACCs and stereoscopic vision on the 3rd and 6th ribs, respectively. Table 2 shows the RMSE of the stereoscopic vision measurement compared to the IR-TRACC measurement results. The stereoscopic vision measurement overall traced the rib deflection with small RMSE, particularly, the RMSE of resultant deflection of both the 3rd and 6th ribs were smaller than 1.0 mm.

Figure 10 shows the comparison between the predicted rib deflections through the 2nd to 6th ribs and the optically measured rib deflections. The stereoscopic vision measurement results for the 2nd and 4th ribs were interrupted because of the obstruction of the camera view during the experiment. The noise occurred in the Z direction deflection of the 4th rib because the compensation by the ratio of upper and lower measurement results of 3rd rib deflection rose drastically within a short duration. Table 3 shows the RMSE and the differences of the peak resultant deflections of Test 3.

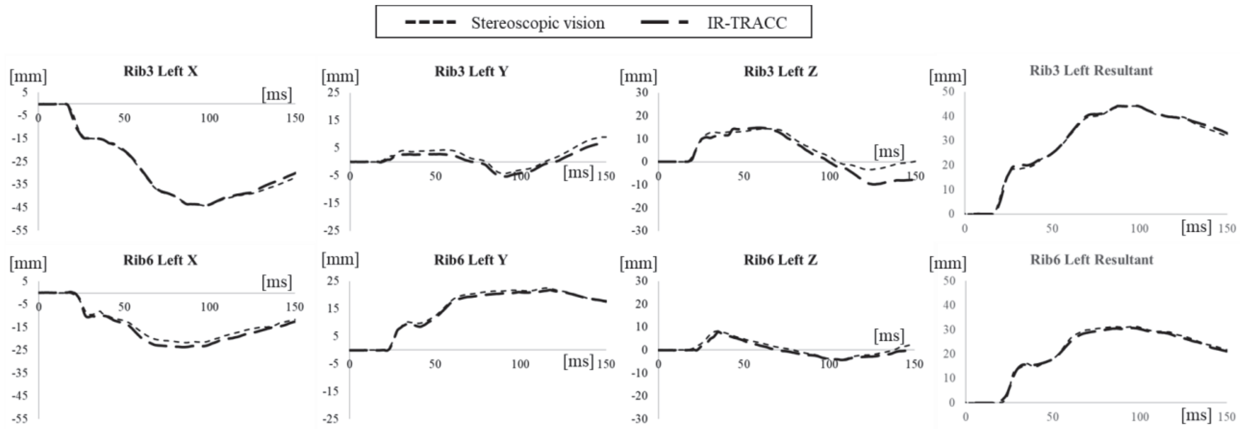


Figure 9. The rib deflection measured by stereoscopic vision compared to that measured by IR-TRACC

Table 2.
The RMSE of the comparison of the stereoscopic vision to the IR-TRACC measurement

		X [mm]	Y [mm]	Z [mm]	Resultant [mm]
Left	**Rib 3**	0.8	1.4	3.5	0.7
	Rib 6	1.7	0.7	1.0	0.8

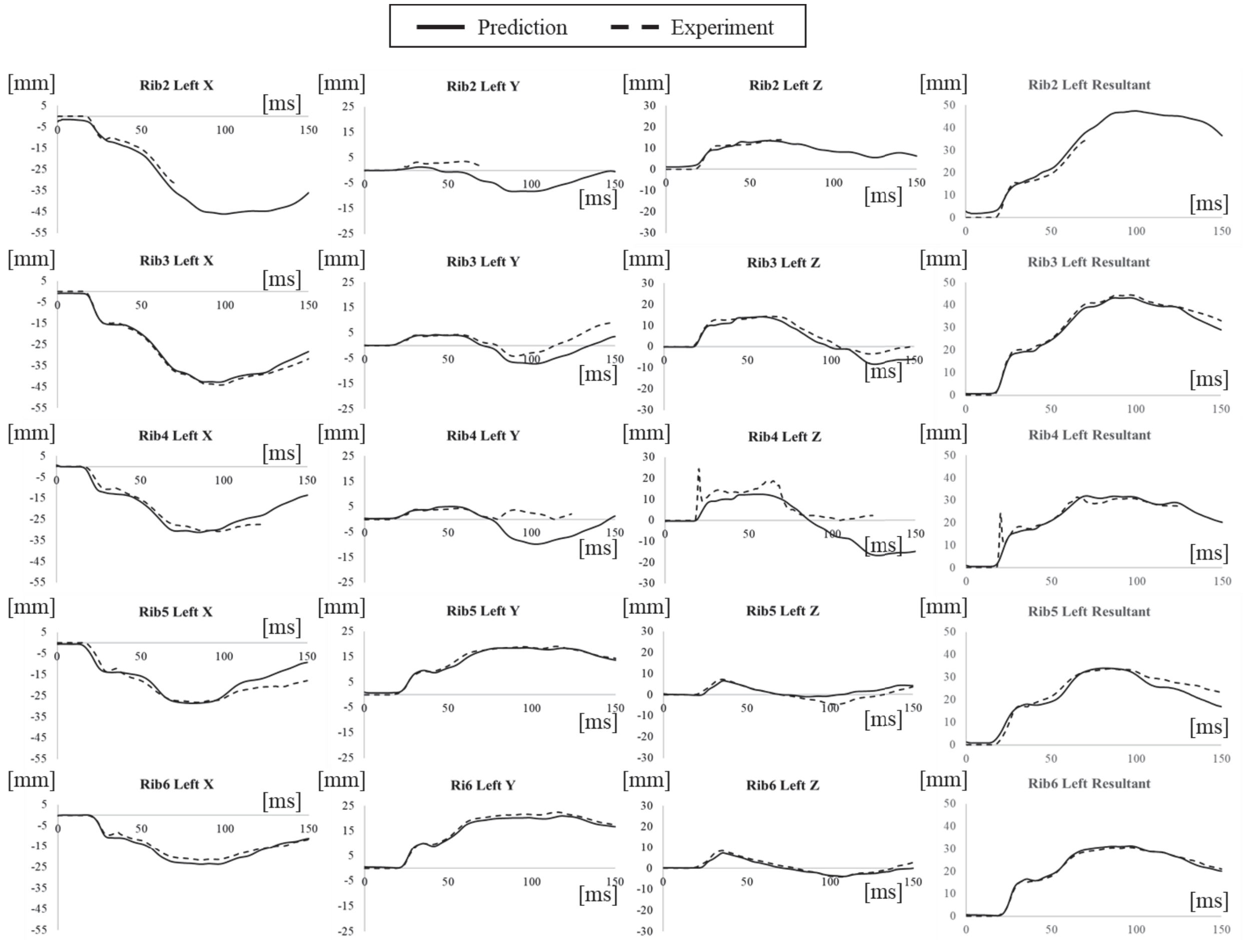


Figure 10. Comparison of predicted deflection by the DNN model and stereoscopic vision

Table 3.
The RMSE and the differences of the peak resultant deflections of Test 3

			X [mm]	Y [mm]	Z [mm]	Resultant [mm]	Differences of Peak Res. between Pred. and Exp. [mm]
		Rib 2	2.3	2.8	1.0	2.1	—
		Rib 3	1.4	3.5	2.8	1.5	1.0
Test 3	**Left**	**Rib 4**	2.1	6.1	6.9	2.3	0.0
		Rib 5	3.6	0.6	2.0	2.7	-0.3
		Rib 6	1.6	1.1	0.9	0.7	-0.6

DISCUSSION

The predicted time-historical rib deflections by DNN model showed errors with smaller RMSE than average 1.5 mm, validated by means of measured deflections by IR-TRACC and stereoscopic vision. Table 3 showed that the differences of predicted and experimental peak resultant deflection in Test 3 indicated smaller error than ±1.0 mm also in ribs without IR-TRACCs, although the model was trained by a small number of datasets. The reason for such small errors was assumed to be that the FE model for the training data was validated sufficiently to represent the actual mechanical properties among the ribs with and without IR-TRACCs and that information was included in the DOE of the 56 training simulation results. In addition, the THOR-ATD bib was independently validated under three-point bending loading conditions [17]. Owing to these validations, the mechanical property of the FE simulation was assumed to represent the actual properties with high accuracy, resulting in the decrease of DNN prediction error.

Although resultant deflections were predicted with small errors, Y direction deflection in the upper ribs showed larger errors than those of X and Z direction deflections. These trends were indicated consistently within the test results of Tests 1, 2, and 3. This was considered to be due to the influence of some of the boundary conditions of the thoracic calibration test for THOR-ATD, even though the THOR-ATD FE model overall represented well the experimental results with high accuracy. The thoracic calibration was conducted by horizontally impacting the probe on the thorax of THOR-ATD vertically seated on the test table. The response curve of force and deflection was confirmed to fall into the corridor [17]. As shown in Figure 1, since the tips of the ribs on the THOR-ATD are downward in order to more exactly represent human ribs, the directions of deflection of the impacted ribs are dominantly the X and Z directions. Consequently, Y direction deflection was relatively smaller than those of the other two directions. Hence, the validation of Y deflection would be insufficient by the calibration test, whereas the validation accuracy of X and Z deflection was improved by fitting within the response curve corridor. Furthermore, the amount of Y deflection was smaller than those of the other two directions, resulting in increasing the sensitivity against the error, which might decrease the validation accuracy. On the other hand, the oblique impact calibration test was conducted on the lower part of the thorax. The test mode may improve the accuracy of Y direction deflection of the ribs in the lower part.

The RMSE of the right 3rd rib deflection was more than twice those of the three predictions. Those results may be due to the influence of the seat belt path, which passes from the right shoulder to the left abdomen as shown in Figure 11. The seat belt passes on the surface of left 3rd rib and directly push into the thorax. On the other hand, seat belt force on the right thorax was transmitted to the 3rd rib indirectly through the 1st and 2nd ribs. Maatouki et al. reported that the validation accuracy of the 3rd rib deflection was small when the probe impacted on an upper surface of the thorax around the 1st and 2nd ribs compared to when impacting directly on the 3rd rib [17]. Furthermore, although the THOR-ATD FE model was validated by the sled experiment using a seat belt, the main focus of the test was on the deflection of the ribs right under the seatbelt path where the maximum rib deflection mainly occurred. Therefore, the priority of the validation accuracy improvement was low on the ribs far from the seatbelt. For these reasons, the validation accuracy of the right 3rd rib was inferior to the other three ribs with IR-TRACCs and the prediction accuracy was also smaller than those of the other three ribs. Whereas the right 6th rib was also far from seat belt path, the influence of the seatbelt path may be small because the amount of the deflection itself was small.

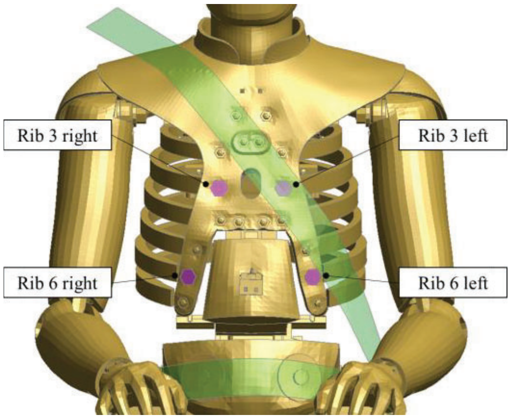


Figure 11. Illustration showing the location of seat belt path and IR-TRACC

As limitations, the prediction accuracy of a specific direction was small due to the influence of the validation test condition and the seat belt path. Some of the ribs indicated an RMSE smaller than 1.0 mm. However, it would be necessary to improve the accuracy in order to replace a physical measurement device used for the assessment test. The additional learning data under various boundary conditions would be necessary in order to improve the prediction accuracy. Furthermore, this study validated only the 2nd to 6th ribs on the left side, and the other ribs also need to be validated.

CONCLUSIONS

The DNN model constructed by the FE simulation predicted all time-historical rib deflections from the four IR-TRACCs and the sled velocity as input data. When the well-validated FE simulation results were used as training data, the DNN was able to learn the physical characteristics, which could predict the time-historical deflection of the physical THOR-ATD. These results indicated the potential for application of the artificial intelligence model as an alternative to the measurement devices.

REFERENCES

[1] Kent, R.W., Henary, B., Matsuoka, F., 2005. "On the fatal crash experience of older drivers." Annu Proc Assoc Adv Automot Med. 49: 371–391.

[2] Kent, R.W., Woods, W., Bostrom, O., 2008. "Fatality risk and the presence of rib fractures." Ann Adv Automot Med. 52: 73–84.

[3] National Highway Traffic Safety Administration. 2017. "Traffic Safety Facts 2015." DOT HS 812 372, 1-5, Available at https://crashstats.nhtsa.dot.gov/#/DocumentTypeList/12. Accessed November 23, 2022

[4] Stawicki, S. P., Grossman M. D., et al., 2004. "Rib Fractures in the Elderly: A Marker of Injury Severity." Journal of the American Geriatrics Society. 52(5): 805-808.

[5] Bergeron, E., Lavoie, A., et al., 2003. "Elderly trauma patients with rib fractures are at greater risk of death and pneumonia." The Journal of Trauma: Injury, Infection, and Critical Care, 54(3): 478-485

[6] Bulger, E. M., Arneson, M. A., Mock, C. N. Jurkovich, G. J., 2000. "Rib Fractures in the Elderly." The Journal of Trauma: Injury, Infection, and Critical Care, 48(6): 1040-1047

[7] Kent, R.W., Patrie, J., Poteau, F., 2003. "Development of an age-dependent thoracic injury criterion for frontal impact restrain loading." The Proceedings of 18th International Technical Conference on Enhanced Safety of Vehicles (ESV), Nagoya, Japan; 12

[8] Craig, M., Parent, D., Lee, E., Rudd, R., Takhounts, E., Hasija, V., 2020. "Injury Criteria for the THOR 50th Male ATD.", Available at https://lindseyresearch.com/wp-content/uploads/2021/10/NHTSA-2020-0032-0005-Injury-Criteria-for-the-THOR-50th-Male-ATD.pdf. Accessed November 23, 2022

[9] Poplin, G.S., McMurry, T.L., Forman, J. L., Ash, J., Parent, D. P., Craig, M. J., Song, E., Kent, R., Shaw, G., Crandall, J., 2017. "Development of thoracic injury risk functions for the THOR ATD.", Accident Analysis and Prevention, 106: 122-130

[10] Kawabuchi, T., Yasuhiro, D., 2019. "Evaluation of Thoracic Deflection Criteria in Frontal Collision Using Thoracic Impactor Simulation with Human Body FE Model.", The Proceedings of 26th International Technical Conference on Enhanced Safety of Vehicles (ESV), Einthoven, Netherland

[11] Guo, X., Li, W., Lorio, F., 2016., "Convolutional Neural Networks for Steady Flow Approximation.", The Proceedings of 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining: 481-490

[12] Ito, O., Shiraishi, J., "A Study of Pedestrian Head Protection CAE Accuracy Improvement Using Deep Learning (Japanese)", Paper presented at: Society of Automotive Engineers of Japan (JSAE), paper #20206294

[13] European Commission Eighth Framework Programme Horizon 2020, 2017. "Safety Enhanced Innovations for Older Road Users", Deliverable 2.5a

[14] Humanetics Innovative Solutions, 2016. "THOR-50th Metric V1.3.2 LS-DYNA Model Technical Report User's Manual."

[15] Ronneberger, O., Fischer, P., Brox, T., 2015. „U-Net: Convolutional Networks for Biomedical Image Segmentation.", ArXiv 1505.04597

[16] Crandall, J., 2013. "Thor Metric SD3 Shoulder Advanced Frontal Crash Test Dummy.", DTNH22-09-H-00247

[17] Maatouki, I., Fu, S., Zhou, Z., 2018. „Latest FE Model Development of THOR-50M Crash Test Dummy", 15th International LS-DYNA Users Conference

Appendix A

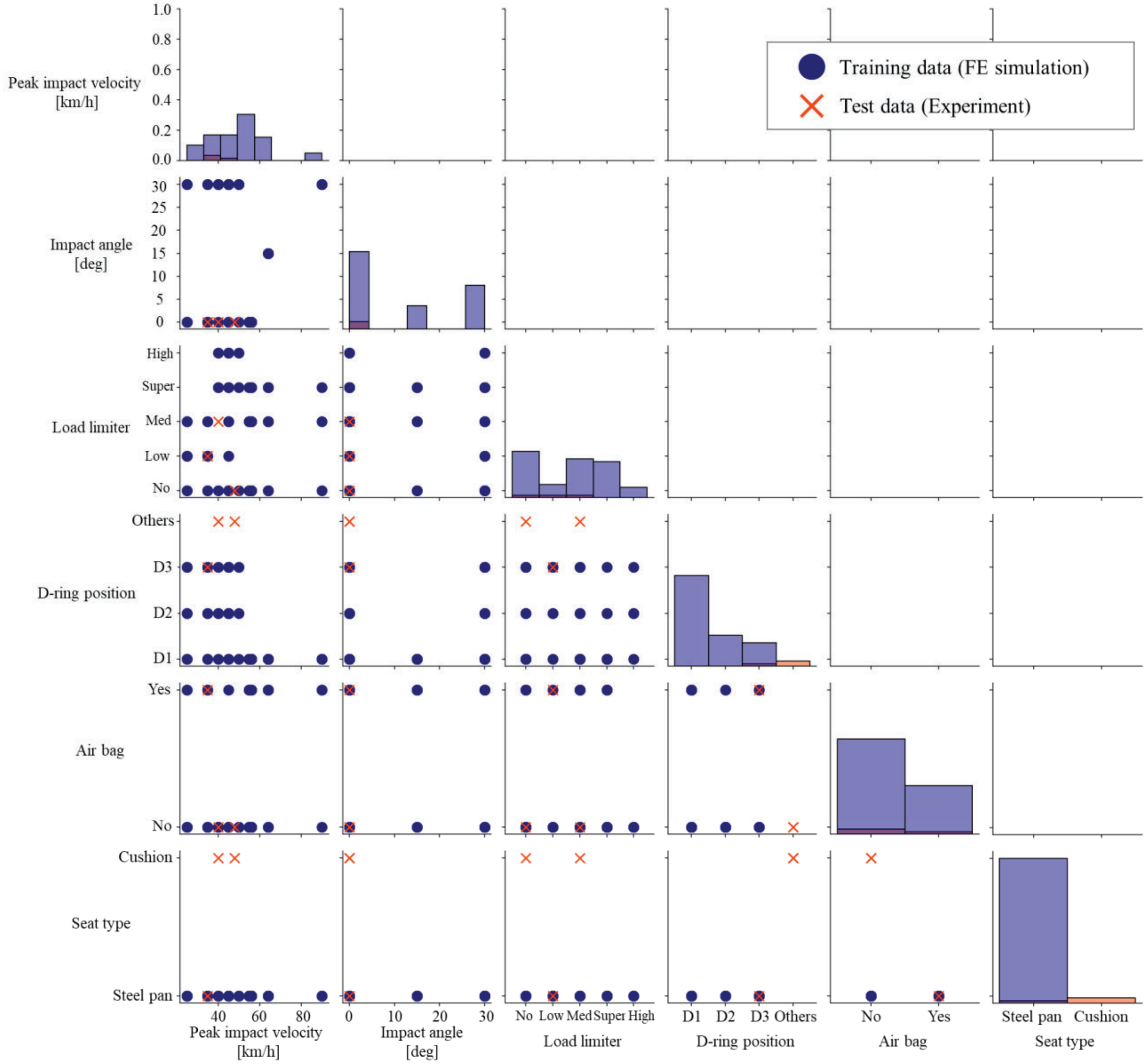
Table A1.

The load case of the training and test data

Test No.	Peak impact velocity [km/h]	Impact angle [deg]	Load limiter	D-Ring position	Airbag	Seat type
1	25	0	No	D1	No	Steel pan
2	25	0	Low	D2	No	Steel pan
3	25	30	Med	D3	Yes	Steel pan
4	35	0	No	D2	No	Steel pan
5	35	0	Low	D3	Yes	Steel pan
6	35	30	Med	D1	No	Steel pan
7	45	0	Med	D2	No	Steel pan
8	45	30	No	D3	No	Steel pan
9	25	0	Med	D3	No	Steel pan
10	25	0	No	D1	Yes	Steel pan
11	25	30	Low	D2	No	Steel pan
12	35	0	Low	D3	No	Steel pan
13	35	0	Med	D1	No	Steel pan
14	35	30	No	D2	Yes	Steel pan
15	45	0	Med	D2	Yes	Steel pan
16	45	0	No	D3	No	Steel pan
17	45	30	Low	D1	No	Steel pan
18	45	0	Super	D2	No	Steel pan
19	40	0	Super	D2	No	Steel pan
20	50	0	High	D2	No	Steel pan
21	45	0	High	D2	No	Steel pan
22	40	30	No	D1	No	Steel pan
23	40	30	Super	D2	No	Steel pan
24	40	30	High	D3	No	Steel pan
25	45	30	Super	D3	No	Steel pan
26	45	30	High	D1	No	Steel pan
27	45	30	No	D2	No	Steel pan
28	50	30	No	D3	No	Steel pan
29	50	30	Super	D1	No	Steel pan

Test No.	Peak impact velocity [km/h]	Impact angle [deg]	Load limiter	D-Ring position	Airbag	Seat type
30	56	0	No	D1	Yes	Steel pan
31	56	0	Med	D1	No	Steel pan
32	56	0	Super	D1	No	Steel pan
33	45	30	No	D1	No	Steel pan
34	45	30	Super	D1	Yes	Steel pan
35	45	30	Med	D1	Yes	Steel pan
36	64	15	Super	D1	No	Steel pan
37	64	15	No	D1	Yes	Steel pan
38	64	15	Med	D1	Yes	Steel pan
39	56	0	Med	D1	No	Steel pan
40	56	0	Super	D1	No	Steel pan
41	56	0	No	D1	Yes	Steel pan
42	64	15	Super	D1	Yes	Steel pan
43	64	15	No	D1	No	Steel pan
44	64	15	Med	D1	Yes	Steel pan
45	56	0	Med	D1	No	Steel pan
46	56	0	Super	D1	No	Steel pan
47	56	0	No	D1	Yes	Steel pan
48	64	15	Super	D1	Yes	Steel pan
49	64	15	No	D1	No	Steel pan
50	64	15	Med	D1	Yes	Steel pan
51	55	0	No	D1	No	Steel pan
52	55	0	No	D1	Yes	Steel pan
53	55	0	Med	D1	Yes	Steel pan
54	55	0	Super	D1	No	Steel pan
55	55	0	Super	D1	Yes	Steel pan
56	55	0	Med	D1	No	Steel pan
Test 1	35	0	Low	D3	Yes	Steel pan
Test 2	48	0	No	Others	No	Cushion
Test 3	40	0	Med	Others	No	Cushion

Appendix B
Figure B1.
Visualized diagram of DOE



A CONCEPT TO SUPPORT AI MODELS BY USING ONTOLOGIES - PRESENTED ON THE BASIS OF GERMAN TECHNICAL SPECIFICATIONS FOR LANE MARKINGS

Maximilian Grabowski
Federal Highway Research Institute
Germany

Ya Wang
Fraunhofer Institute for Open Communication Systems
Germany

Paper Number 23-0265

ABSTRACT

Artificial Intelligence (AI) and Machine Learning (ML) deliver promising approaches to the development of assisted as well as automated and autonomous driving [1] technologies. However, learning all possible traffic situations and outcomes is almost not feasible. Furthermore, machine learning-based models are usually regarded as a black box and we cannot trace their decisions for a certain behavior. To counteract this, we propose an ontology-based model, which integrates normative knowledge, to support the decision making of the AI for automated and autonomous vehicles. Since traffic rules and laws are explicitly defined in the model, we can easily track any derived decisions, eliminating the necessity of learning all possible traffic situations. We formalize the German Technical Specifications on Lane Markings into an ontology for a better representation of the traffic environment and thus improve the situational awareness of automated and autonomous vehicles. Additionally, the reasoning capacity of an ontology based-model allows for deriving concepts in multiple ways, which can serve as redundant information about lane and lane markings to enhance the understanding of the traffic situation. Finally, in contrast to learning-based models, our transparent ontology-based model allows for the validation and verification of automated and autonomous systems and vehicles.

INTRODUCTION

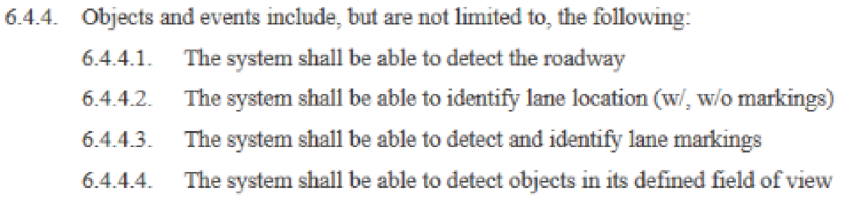
Advanced driver-assistance systems (ADAS) are usually programmed on rule-based algorithms. With the increasing level of automation, the requirements for these systems become more complex, especially for the automated driving systems (ADS). These ADS must comply with rules and laws, consider guidelines, mathematical and physical laws, expert, and world knowledge. This increasing complexity poses a huge challenge for the automotive industry, because there are too many hand-crafted rules that need be taken into account while programming. To this end, data-driven AI and ML algorithms are a very promising tool to solve the challenge of the high complexity. With the development of various advanced sensors and communication systems, more and more data is generated in traffic and available for processing with learning-based models. In this field, driving data recorded with cameras, radar, and lidar are essential for the development of functional and safe systems. These sensors can record, depending on the type, from 1 MB/s up to 500 MB/s, which translates to about 6 TB per day for a setup of assistance Level 2 or even 100 TB per day for a setup of autonomous driving at level 4 and 5 [2]. This amount of data and the increasing computational power allows for the application of data-driven AI models. For the use of AI models, we need to consider two types of applications. There are non-safety-critical and safety-critical applications. The former ones include speech processing, virtual assistants, chatbots, and search engines, which has relatively high tolerance to errors and misclassifications. The latter include diagnostics in medicine and the automated driving in the automotive industry, where a misclassification can result in severe injuries or even death. In the case of automated and autonomous driving, the machine learning-based models can be used for the object classification of the perceived surrounding of the automated and autonomous vehicle (AV), for the recognition of driving scenarios and for the subsequent trajectory prediction and planning of the AV. All these are safety-critical tasks that must perform properly and be robust in their abilities, which highlights the importance of the validation and verification of AI models used in automated and autonomous systems. A shortcoming of learning-based models is that they cannot learn all possible rules, laws, and interactions on the street to achieve higher accuracy. Additionally, using imitation learning [3], AI systems are prone to learn non-rule conforming behavior. An example of learning wrong rules is keeping a minimal distance when a car is passing a bicyclist. According to the German traffic regulations, in urban areas the passing vehicle must keep a distance of at least 1,5 m to the bicyclist and in non-urban areas the passing vehicle must keep a distance of at least 2,0 m (Sec. 5, para. 4, StVO) [4]. This law is fairly new and not many drivers know this new distance or they just do not follow the rules. From this, we can never learn rule compliant behavior for safety concerns. A favored idea, to overcome the shortcomings of solely data-driven AI models, is the proper combination of rule-based algorithms and data-driven algorithms to enhance their advantages and eliminate disadvantages. In the field of automotive engineering, a lot of research is being performed to aggregate, consolidate, and harmonize different terms for a unified traffic scene representation with respect to the driving safety of automated and autonomous vehicles [5]. For the representation of a complex structured traffic environment, Scholtes et al. [6] proposed the 6 Layer Model. Maierhofer et al. [7] formalized partially machine-readable German traffic regulations in temporal logic. Bermejo et al. [8] demonstrated that the ontology-based representation of traffic scenes can enhance situational comprehension on the traffic roads, improve traffic safety, and support a centralized traffic management system. Bagschik et al. [9] proposed an ontology to assist experts in creating scenes based on formalized knowledge covering a wide range of scenarios. Other research focused on deriving rule compliant behavior of road users at complex road intersections, where an ontology was used to infer which agent has the right of way [10, 11].

In this work, we propose a concept to support AI models by formalizing the German Technical Specifications on Lane Markings within an ontology. The German Road and Transportation Research Association (FGSV) defines technical standards and specifications related to road and transportation. Part of these standards and specifications pertain to the definitions of lane markings [12]. The knowledge from these documents is formalized into an ontology by creating concepts and relations between concepts. It can then be used in AI algorithms to support the entire system. With the support of the lane marking ontology, the AV will be able to reason more information about lane markings and thus enhance its understanding on the traffic situation, such as inferring traffic areas of lane markings and identifying the type of lane markings when accurate sensor data is missing. This information is crucial for subsequent trajectory prediction and planning tasks to ensure that the AV derives rule compliant behavior.

ONTOLOGY

The word ”ontology” has different meanings depending on the community. In philosophy, the branch that deals with the nature and the structure of reality, ontology is the study of attributes that belong to things due to their nature [13]. In Computer Science, ontology takes a different meaning, which is however related and based on the philosophical meaning. Gruber et al. [14] defines the computational meaning of an ontology as the explicit specification of a conceptualization of a domain knowledge. This means that an ontology describes concepts and relations that can exist between concepts [14]. Advantages of an ontology are the efficient exchange of information due to sharing and reusing formalized knowledge and that they are human and machine readable, as it clearly assigns semantics to unambiguous concepts represented by a set of unique symbols [13]. Humans are able to directly identify concepts, its hierarchical structure, axioms, and rules. Machines understand the concepts and relationships due to the unique symbols, hence they can handle the symbols with logic using computer software [9]. For a better representation of knowledge in an ontology, RDF Schema, Ontology Web Language (OWL) and If-Then rules are often used [15]. Ontologies usually comprise a terminological box (T-Box) and an assertional box (A-Box) [16]. The T-Box defines concepts of the ontology, where these concepts are called classes that can have data type properties or constraints, relationships between classes, and axioms and rules [9]. The A-Box describes specific instances of classes called individuals [9], which are taken for certain situations. Modelling and verification of the ontology can be done in a software like protege [17, 18, 19].

In this work, an ontology for lane markings is created using the formalized knowledge from relevant national documents. We demonstrate that the structure of knowledge in the corresponding documents can easily be fitted into the hierarchical structure of ontologies, furthermore the relational information can be used to capture more important knowledge on lane markings. Additionally, the created ontology on lane markings is integrated into a base ontology



6.4.4. Objects and events include, but are not limited to, the following:

6.4.4.1. The system shall be able to detect the roadway

6.4.4.2. The system shall be able to identify lane location (w/, w/o markings)

6.4.4.3. The system shall be able to detect and identify lane markings

6.4.4.4. The system shall be able to detect objects in its defined field of view

Figure 1. Guidelines for requirements for automated and autonomous vehicles proposed by the Informal Working FRAV. Taken from [20]

on the traffic domain, the combined one is further expanded to include missing concepts of the autonomous driving domain for example use cases, and it is tested in a reasoning software in combination with exemplary traffic rules to show benefits of an extended ontology.

FORMALIZATION OF NORMATIVE KNOWLEDGE

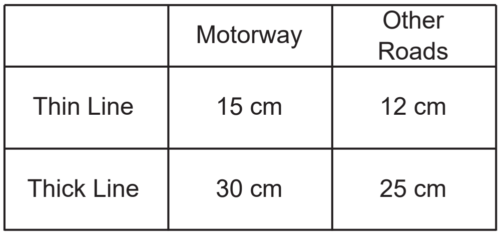
In this section, we present the legal framework for the type-approval of vehicles, that includes automated and autonomous vehicles, whether they are based on AI models or not. This framework leads to relevant documents on international and national level. On an international level, we show the working document of the UNECE working group for Functional Requirements for Automated Vehicles (FRAV), which defines the requirements for automated vehicles as a guideline document. On the national level, we discuss the German Technical Specifications on Lane Markings from the German Road and Transportation Research Association (FGSV), which is the main contributor of expert knowledge for this work. After extraction, we present the structure of the knowledge and demonstrate how it can be translated into an ontology, which is specialized in the domain of lane markings. We take this ontology and integrate it into a more general one, which functions as a basis. This basis is the ASAM OpenXOntology, which offers formalized knowledge on the general domain of traffic and thus it is easily extended with knowledge on lane markings.

Knowledge Sources

On an international level, there are activities to advance the regulation of automated and autonomous vehicles. The responsibility of automated and autonomous vehicles lies within the Working Party on Automated/Autonomous and Connected Vehicles (GRVA), which prepares draft regulations and guidance documents. The GRVA set out a mandate that a subgroup must formulate and develop functional requirements for automated/autonomous vehicles, which resulted in the formation of the Informal Working Group (IWG) on Functional Requirements for Automated/Autonomous Vehicles (FRAV). Development shall be captured in a working document, which functions as a guideline document [20]. Guideline documents of this kind do not have regulatory character, however they will be the basis for upcoming UN Regulations (UN-Rs) or UN Global Technical Regulations (GTRs), which must be adhered to when type-approving new vehicles. Hence, it makes sense to to consider the guidelines created by FRAV during the design process of new automated and autonomous functions and vehicles.

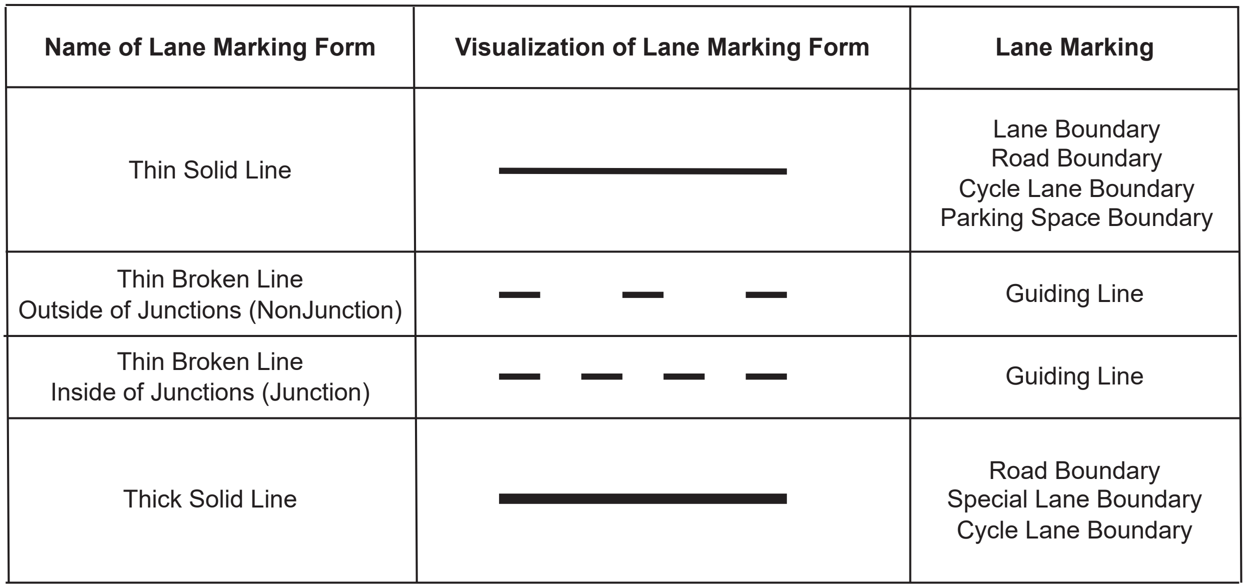
There are numerous requirements formulated in the document, however these requirements have different areas of importance. Fig. 1 shows a selection of requirements for the Object and Event Detection and Response (OEDR). Here, the first requirement is the ability of the system to detect the roadway. A relevant question for this requirement is ”What is a roadway?”. There are different possibilities to identify the roadway for an automated and autonomous vehicle, however, we come to the conclusion that one major aspect for the identification of the roadway are lane markings. Additionally, the detection and identification are also requirements formulated by FRAV. Hence, if we are able to correctly detect and identify lane markings, and later handle them properly, we are one step closer to safe and transparent autonomous driving. Even if the general idea of lane markings is the same in most countries, there are many different types, shapes, and forms of lane markings with different meanings, which have an effect while following traffic regulations. Thus we are taking lane markings as a use case for this work to show how this kind of knowledge can be formalized into ontologies for the development of automated and autonomous vehicles.

Here, we concentrate on the German Technical Specifications for Lane Markings (RMS-1) [12] as it handles all relevant traffic areas (urban, non-urban, and motorway) and the general knowledge on type, dimension, and form of



	Motorway	Other Roads
Thin Line	15 cm	12 cm
Thick Line	30 cm	25 cm

Figure 2. Width of longitudinal lane markings for thin and for thick lines and for motorways and other roads (urban and non-urban). Adapted from Ref. [12]



Name of Lane Marking Form	Visualization of Lane Marking Form	Lane Marking
Thin Solid Line	———	Lane Boundary Road Boundary Cycle Lane Boundary Parking Space Boundary
Thin Broken Line Outside of Junctions (NonJunction)	— — —	Guiding Line
Thin Broken Line Inside of Junctions (Junction)	— — — —	Guiding Line
Thick Solid Line	▬▬▬	Road Boundary Special Lane Boundary Cycle Lane Boundary

Figure 3. Excerpt of the table that shows the name of lane marking, the visualization of the lane marking form, and the lane marking type for longitudinal lane markings. Adapted from Ref. [12]

standard lane markings. Further knowledge appears in the form of proper definitions, area of appearance, and conditions for specific lane markings. The document offers a reliable source of knowledge that can easily be extracted and then properly formalized.

Formalization of Knowledge Sources into an Ontology

Before we look at the structure in more detail, we note, that it is stated, in the beginning of the document, that the construction authority must adhere to these technical specifications and larger changes to these specifications must be authorized by the supreme traffic authority. However, these changes cannot alter regulatory law.

The first important section of the Technical Specification is the chapter *Dimension of Lane Markings*. From this chapter, we can extract a significant part of the overall knowledge needed for the ontology. Relevant knowledge is well-structured in tables and easily readable. Generally, we can distinguish between longitudinal markings, markings for restricted areas, markings for stopping and parking prohibition, perpendicular markings, arrows, and miscellaneous markings. Focus will be put on longitudinal and perpendicular markings. For the longitudinal markings, there is a very important distinction between the width of longitudinal lane markings on motorways and on other roads (urban and non-urban regions) and a distinction between thin and thick lane markings, which can be seen in Fig. 2. Then, there are tables, that list the name, visualization of the lane marking form, and the lane marking type of possible lane markings. These can be found for all the aforementioned classes of lane markings (see Fig. 3. The name of the lane marking (left column) describes the basic form in a structured manner, where we obtain the information of the width of the line (thin or thick), structure (solid or broken), number of lines (single or double), and if the line

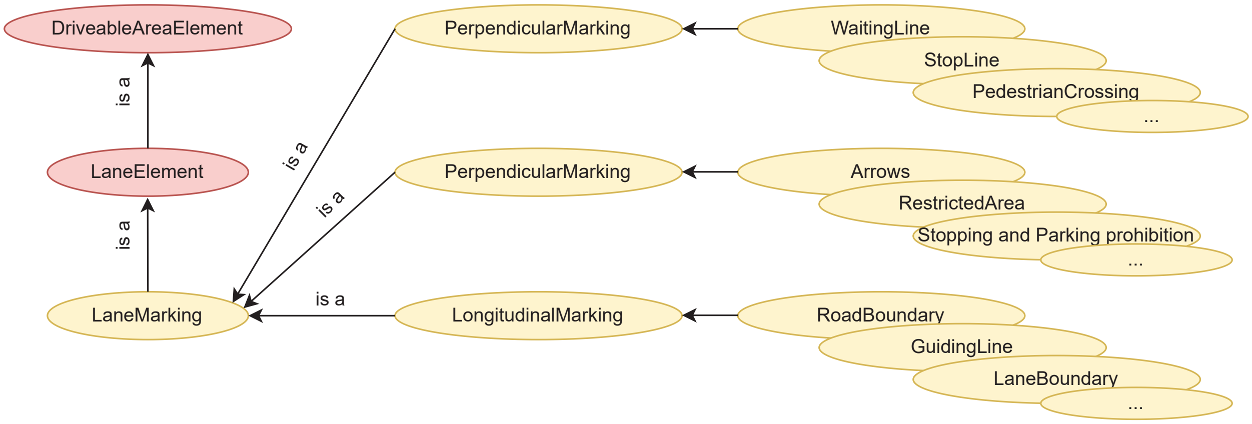


Figure 4. Hierarchical structure of the lane markings type.

is broken in what ration the dashed lines appear (2:1, 1:1, 1:2). The visualization of the form (center column) reflects the name in a visual form, which is useful knowledge for the perception task of an autonomous vehicle. In the last column, there is the lane marking type, which gives a specific name to the basic form and this name (here called type) is the same as referenced in the traffic regulation. Most assertions of basic form and name to a certain type are unique, hence we have only one basic form for one type, however, some assertions are ambiguous, where one form can have several types and some types can have several forms depending on where they appear. In a further table for longitudinal markings, the ratios for the broken lines are defined with their lengths for specific situations, but this information was not in the focus in this work.

Following sections filter lane marking types by their area of appearance regarding junctions and non-junctions and their conditions for appearance. A junction is usually a crossing, where often different types of lane markings appear compared to non-junction areas. For example, solid roadway boundaries often change to broken roadway boundaries with a certain ratio to signify that they can be crossed by traffic. Conditions of appearance are i.e. the existence of roadway boundaries which are usually depicted by solid lines at the sides of the roadway, however in urban areas these solid lines can be omitted, if the sides of a roadway are clearly defined by the architecture like a curbside. We are focusing less on the latter in this work.

While extracting the knowledge, we see that the structure of the information in the Technical Specification is generally hierarchical (see Fig. 4). There are *Lane Markings*, which can be considered a high-level term as this term generally comprises all lane markings. These *Lane Markings* can be grouped into *Longitudinal Markings*, *Perpendicular Markings*, and *Miscellaneous Markings*, where the last comprises markings for restricted areas, markings for stopping and parking prohibition, arrows, and miscellaneous markings. All of these types of markings can be further split into definitive markings, i.e. *Road Boundary* or *Guiding Line*. This kind of hierarchical structure can be set up for 3 different categories (see Fig. 5. The first is the aforementioned branch of *Lane Markings*, which defines the type of lane marking and has specific rules attached to it. The second is the branch of *Lane Marking Form*, which depicts the exact form which can be seen on the street. This branch combines the left and center column of Fig. 3. The last is the *Traffic Area* branch, which defines the traffic areas in which the Lane Markings can appear. In addition to the hierarchical structure, nodes and leaves from each branch have links to other branches, thus relationships can be established. These well-defined concepts, the hierarchical structure, and the relations between concepts makes this knowledge perfectly suitable to be formalized as an ontology. We use protege [18, 19] as a software to construct the ontology, as this software is compatible with the OWL2 language and it allows for an easy development of an ontology. Concepts are created and obtain a proper definition that tries to mirror the knowledge of the Technical Specification. Three branches consisting of the aforementioned high-level concepts *LaneMarking*, *laneMarkingForm*, and *TrafficArea*. A proper ontology is achieved by creating relevant relationships between different concepts. Fig. 6 showcases possible relations between the concept *StopLine* and its related concepts. It shows that *StopLine* can appear in the areas *Urban*, *Non-Urban*, and in *Junction*, i.e. neither on *Motorway* nor in *Non-Junction* areas. This relation is denoted with corresponding object properties, where the affiliation of *WaitingLine* to *Urban*, *Non-Urban*, and *Junction* is described with the newly created relation *isPartOf*. The inverse relation that these concepts can contain a possible *WaitingLine* is described with *contains*. Defining the relations in this way, creates the axiom of inversion,

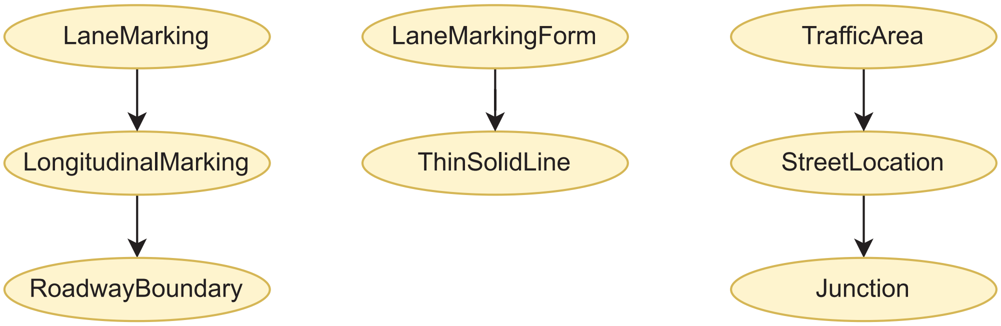


Figure 5. Three branches of identified knowledge from the German Technical Specification on Lane Markings in a hierarchical structure.

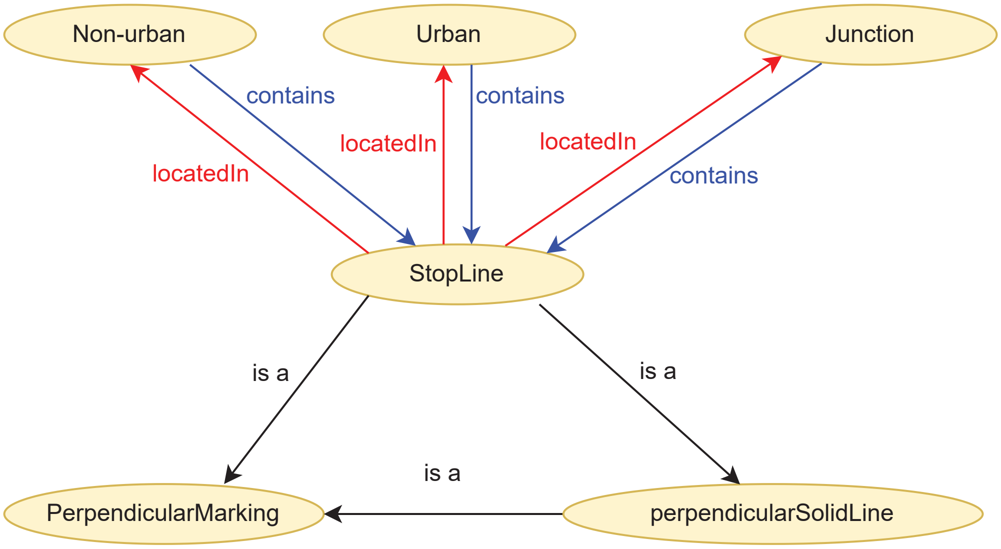


Figure 6. Relations between different concepts of the ontology with StopLine as the center concept.

which is defined by *isPartOf isTheInverseOf contains*. Furthermore, Fig. 6 denotes that it is a *PerpendicularMarking* with a basic form of *perpendicularBrokenLine2To1*, which is defined by the standard relation *is a*.

Integration into ASAM OpenXOntology

A lane markings ontology has been created on the basis of the knowledge from the German Technical Specifications, hence it comprises only knowledge from the specific domain of lane markings. However, to be able to reason in general traffic situations or for an automated or autonomous vehicle to reason in traffic situations, the domain of the ontology must encompass concepts and relations from the general traffic domain and from the automated/autonomous driving domain. A base ontology, that integrates the aforementioned concepts, is needed on which we build upon and integrate our lane markings ontology. The general consensus in the community for ontologies is that you should not create entire ontologies completely from scratch, but rather detect existing ones and identify what can be used as a basis. As a base ontology, we chose the OpenXOntology that was developed by the Association for Standardisation of Automation and Measuring Systems (ASAM). They aim to provide a foundation of common definitions, properties, and relations for central concepts of the ASAM OpenX standards in the domain of road traffic. Choosing the ASAM OpenXOntology has the advantage that it is compatible with OpenDRIVE, OpenSCENARIO, and OpenLABEL standards, that are formats to enable scenario-based testing and that are well-accepted in the automotive engineering community for scenario-based testing. Based on the generality, the ontology is mainly divided into three modules, namely core ontology, domain ontology and application ontology. The core (or upper) ontology is do-

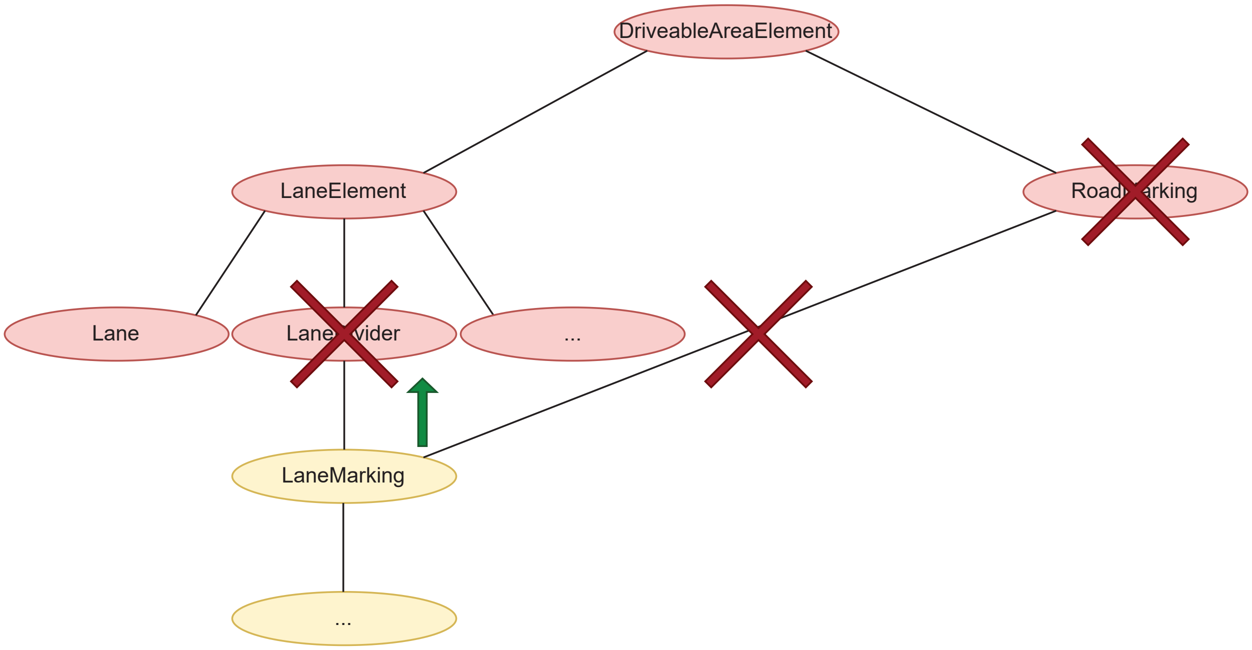


Figure 7. Structural changes of the ASAM OpenXOntology to make the new definition of LaneMarking more suitable.

main independent and describes basic concepts, such as physical objects, states, and events, that are developed based on High-Quality Data Model framework (HQDM). The domain ontology defines central concepts of the road traffic domain consisting of three layers, i.e.,*EnvironmentalCondition*, *RoadTopologyAndTrafficInfrastructure* and *TrafficParticipantAndBehavior*. The application ontology covers the concepts for a specific application, such as *EgoLane* in a simulation application. The OpenXOntology consists of 347 classes, 96 object properties and 2 data properties, and uses OWL as ontology language and Semantic Web Rule Language (SWRL) as rule language.

The methodology to integrate our ontology into the ASAM OpenXOntology is as follows. Take the highest-level concept of our ontology. Identify, if this concept exists in the OpenXOntology by searching the branches in relevant concepts and higher-level concepts. If a corresponding concept exists, compare the names of the concept and see, if it corresponds to your concept name. Should it not be the same, verify, if it needs to be changed and adapt by applying a more suitable name. Additionally, verify, if subclasses need to be adapted. If a corresponding concept does not exist, find a proper higher-level concept and add your concept as a subclass. After adding your concept, add further subclasses to the concept from your own ontology. In our case, the following steps were taken. First, the concept *LaneMarking* is integrated by searching through corresponding concepts of the OpenXOntology. We find the corresponding concept with the same name, which is a subclass of *LaneDivider* and *RoadMarking*. *LaneDivider* is a subclass of *LaneElement* and this concept with *RoadMarking* are subclasses of *DriveableAreaElement*. We adapt *LaneMarking* from our ontology and use *LaneMarking* from the OpenXOntology, however, we restructure the overall structure of the aforementioned concepts, which can be seen in Fig. 7. In the first step, we remove *LaneDivider* as lane markings can be seen as lane dividers when they are longitudinal lane markings, however, our definition of lane markings also includes perpendicular and miscellaneous lane markings, which cannot be seen as lane dividers, hence it is removed and the concept of *LaneMarking* is moved one layer upwards and is now directly a subclass of *LaneElement*. In the second step, we remove *RoadMarking* as a concept from the ontology, because *LaneMarking* is the sole subclass of *RoadMarking* and it serves no other purpose. After integrating *LaneMarking*, we change its subclasses to its created ones *LongitudinalMarking*, *PerpendicularMarking*, and *MiscellaneousMarking* also including each their respective subclasses (see Fig. 4). Second, we integrate the concept *laneMarkingForm* by identifying the concept *laneProperty* as a fitting superclass. Moreover, the new concept fits very well with the already existing sibling classes *laneDimension*, *laneMarkingColor*, and *laneMarkingWidth*. *laneMarkingForm* is further split into the classes *singleLine* and *doubleLine* with their corresponding subclasses (see Fig. 8). Lastly, we integrate the high-level concept *TrafficArea*, which is now considered a subclass of *RoadTopologyAndTrafficInfrastructure*. The sub-

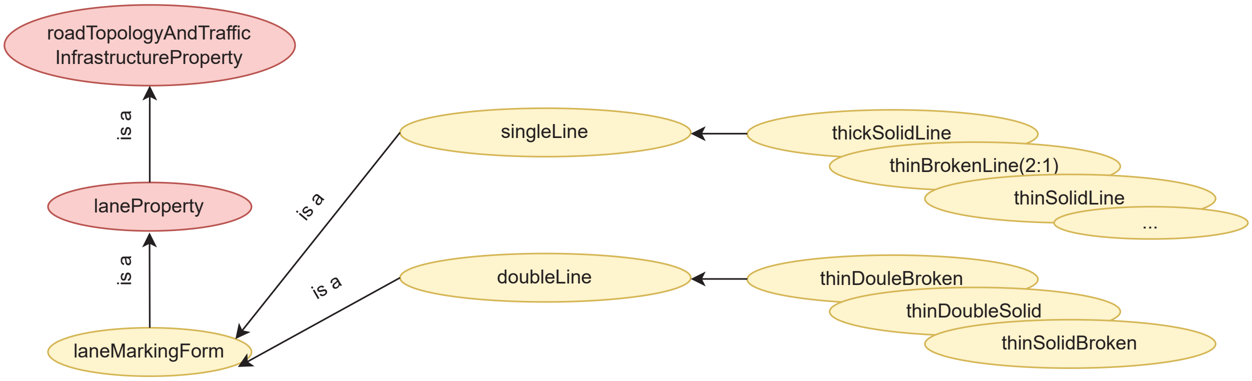


Figure 8. Modified hierarchical structure of the lane marking form.

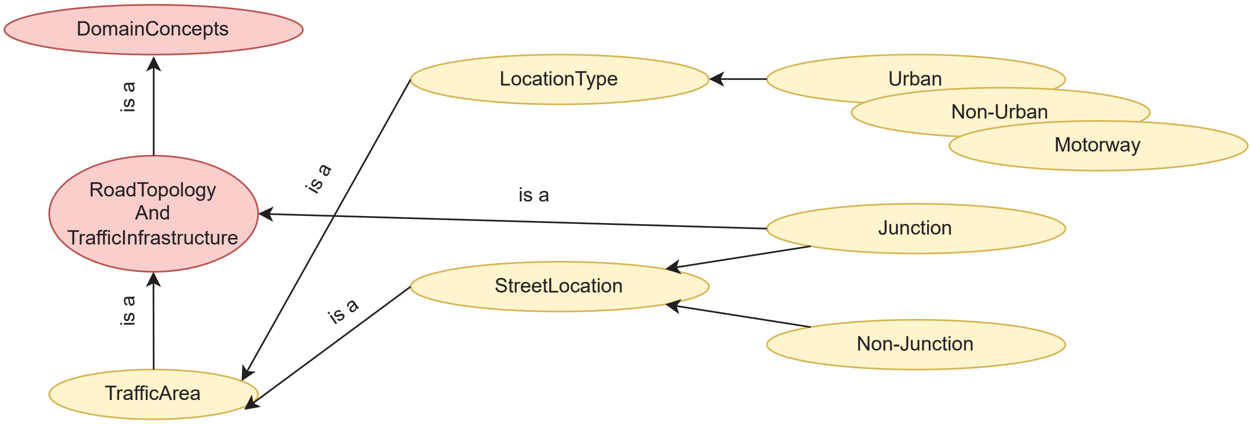


Figure 9. Modified hierarchical structure of the traffic area concept.

classes *LocationType* and *StreetLocation* are added with their respective subclasses (see Fig. 9). Important to note is, that the concept *Junction* was already present in the OpenXOntology as a direct subclass to *RoadTopologyAndTrafficInfrastructure*, *WholeLifeFunctionalSystem*, and *WholeLifeSystem*, which we did not want to remove, hence it has several superclasses in the current version.
After integrating the concepts into the OpenXOntology, relationships between the newly integrated concepts are setup, however no new relationships are created, thus all previous ones are just adopted. In the following chapter, we showcase how the ontology can be used to infer information from the surrounding of the autonomous vehicle. For our specific use cases, we need the concept of *EgoLane*, which is the traffic lane that the ego vehicle is driving on.

DEMONSTRATION ON USE CASES

The normative knowledge formalized with the help of an ontology can support automated and autonomous vehicles not only to make rule compliant decisions [21], but also to improve the confidence of the perceptual results. Although advanced on-board sensors and deep learning-based algorithms have achieved encouraging results in autonomous driving system, they are prone to fail in some edge cases, such as loss of GPS signal and occlusion of lane markings. In this section, we demonstrate that by using formalized normative knowledge the agent is able to correctly detect desired targets by reasoning over perceived data. As shown in Fig. 10, we formalize the German Technical Specification on Lane Markings via an ontology defined in the T-Box and map the Specification into the A-Box. The model takes the real driving data, collected from on-board sensors and computational results of machine learning-based models, as inputs and maps these onto the A-box via the ontology. On the top of the knowledge base consisting of A-Box and T-Box, we build a reasoner that executes queries to answer the questions about the traffic scenarios. We illustrate two use cases, which frequently happen in the real driving environment, to demonstrate the functionality of our model. For implementation, we edit and manage our ontology in protégé [18, 19], formalize our inference rules using SWRL [22], formulate queries using SPARQL [23], and use Stardogs [24] as the knowledge graph platform to store our database and to derive new facts.

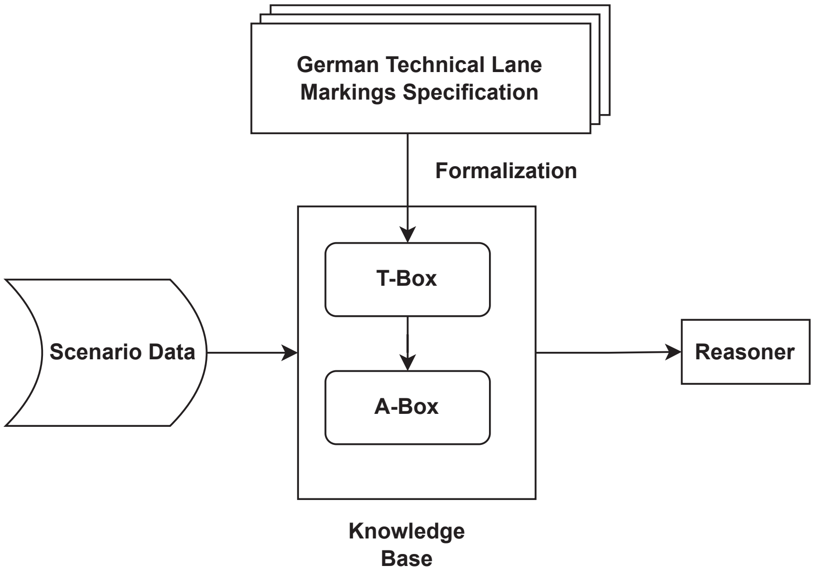


Figure 10. Model architecture consisting of T-Box, A-Box, Reasoner and German Technical Specifications on Lane Markings.

Use Case 1 - Determining the General Traffic Areas

Correct identification of the traffic area in which the current ego vehicle is located is a prerequisite for the proper application of corresponding traffic rules. When losing signals from onboard sensors, such as GNSS and IMU, the detected lane markings can be used as an indicator of the traffic area. According to the German Technical Specification on Lane Markings, we formalize the possible traffic areas for stop line and thin broken lines with the patterns of 1 m to 1 m and 1 m to 2 m in the A-Box as follows:

```
:StopLine :locatedIn :Junction, :Urban, :NonUrban
:thinBrokenLine1To1 :locatedIn :NonUrban, :Urban, :Junction
:thinBrokenLine1To2 :locatedIn :NonUrban, :Urban, :Motorway,:NonJunction
```

After uploading these facts together with the T-Box, we request our knowledge base to answer the following queries in SWRL:

- **Query 1.1:** Retrieve all possible traffic areas that contain stop line.
- **Answer 1.1:** Junction, Urban, NonUrban
- **Query 1.2:** Retrieve all possible traffic areas that contain guiding line.
- **Answer 1.2:** Junction, Urban, NonUrban, NonJunction, Motorway

Although we use *:contains* as the keyword to formulate the relation between the traffic area and the lane marking in our queries, the system can still retrieve the results in the first query by using the owl property axiom, i.e., *:contains owl:inverseOf :locatedIn*. In the second query, we ask about the guiding line, which is not explicitly defined in the A-Box. However, the thin broken lines 1 m to 1 m and 1 m to 2 m are defined as the subclasses of guiding line. Thus, the system can infer all five possible traffic areas for guiding line.

Use Case 2 - Determining the Type of Guiding Line using Environmental Context

The pattern of the lane markings varies in traffic scenes, indicating different meanings. For instance, the broken line with the pattern 1 m to 2 m is used for cycle guiding line and 3 m to 6 m is for the urban guiding line. Correctly identifying the guiding line is meaningful to the driving safety, such as keeping safe distance to the cyclist. These two types of lines are not easy to distinguish without any reference, since they have the same ratio of the lengths. However, we utilize the context information to infer the type of the target using our ontology.

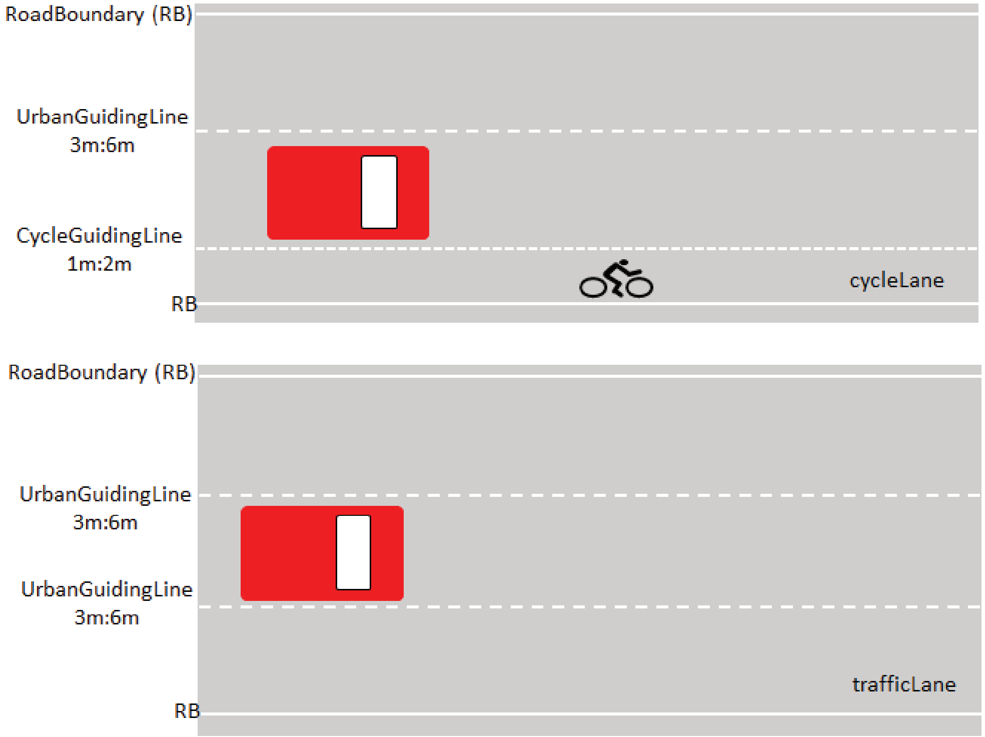


Figure 11. Ego Lanes left connected to the Urban Guiding Lines and right connected to the Cycle Guiding Line (top), and right connected to the Urban Guiding Line (bottom).

We illustrate two traffic scenes as depicted in Fig. 11, in which the top one consists of an ego lane, left connected to the urban guiding line and right connected to the cycle guiding line, and the bottom one is similar except that the ego lane is right connected to the urban guiding line. We formalize our common sense knowledge according to the German Technical Specification on Lane Markings as rules using SWRL as follows:

$$\begin{aligned}&EgoLane(?l1) \wedge LaneMarking(?m1) \wedge LaneMarking(?m2) \wedge CycleLane(?l2)\\&\wedge leftConnectedTo(?l1,?m1) \wedge rightConnectedTo(l1,?m2) \wedge leftConnectedTo(?l2,?m2)\\&\Rightarrow CycleGuidingLine(?m2)\end{aligned} \tag{1}$$

The rule (1) states that, if a cycle lane on the right is detected, then the guiding line left connected to this lane must be a cycle guiding line.

$$\begin{aligned}&EgoLane(?l1) \wedge LaneMarking(?m1) \wedge LaneMarking(?m2) \wedge CylceGuidingLine(?m2)\\&\wedge leftConnectedTo(l1,?m1) \wedge rightConnectedTo(l1,?m2) \wedge leftConnectedTo(?l2,?m2)\\&\Rightarrow CycleLane(?l2)\end{aligned} \tag{2}$$

The rule (2) states that, if a cycle guiding line on the right is detected, then the lane right connected to this line must be a cycle lane.

$$
\begin{aligned}
&EgoLane(?l1) \wedge LaneMarking(?m1) \wedge LaneMarking(?m2) \wedge UrbanGuidingLine(?m2) \\
&\wedge leftConnectedTo(l1, ?m1) \wedge rightConnectedTo(l1, ?m2) \wedge leftConnectedTo(?l2, ?m2) \\
&\Rightarrow TrafficLane(?l2)
\end{aligned} \tag{3}
$$

The rule (3) states that, if an urban guiding line on the right is detected, then the lane left connected to this line must be a traffic lane. When the onboard sensors detect the lane marking on the right with very low confidence and a cycle lane is detected with very high confidence (Figure 11, top), we can query the type of the lane markings as follows:

- **Query 2.1:** Ask if the lane marking right connected to ego lane is cycle guiding line.
- **Answer 2.1:** True

Conversely, if the onboard sensors detect a lane marking with the pattern 1 m to 2 m (Figure 11, top), we can infer the type of this lane marking and the type of lane to the right, since the lane marking with the pattern 1 m to 2 m is assigned to the type of cycle guiding line in the T-Box.

- **Query 2.2:** Ask if the lane marking right connected to the ego lane is the cylce guiding line and the lane right connected to the lane marking is a cycle lane.
- **Answer 2.2:** True

If the the onboard sensors detect a lane marking with the pattern 3 m to 6 m (Figure 11, bottom), we can infer that this line is an urban guiding line and only that the lane on the right is a traffic lane.

- **Query 2.3:** Ask if the lane marking right connected to the ego lane is an urban guiding line and the lane right connected to the lane marking is a traffic lane.
- **Answer 2.3:** True

DISCUSSION

In this work, we showcased the German Technical Specification for Lane Markings as a reliable source of knowledge that can be formalized into an ontology. Knowledge, such as lane marking types, their forms, and their area of appearance were formalized and tested in exemplary use cases, which shows that formalized lane marking specifications can be used to enrich the driving situation. However, we must claim that we do not attempt to create a full ontology with all the aforementioned knowledge but rather to present a conceptual work to demonstrate the effectiveness of the ontology.

The demonstration of the uses cases shows that we can, i.e., infer information on the traffic area by querying possible traffic areas of detected lane markings or determine the type of a lane marking, that can take several lane markings forms, by using further contextual information. The Reasoning times for the first use case are short (below 100 ms) as the reasoner solely uses explicit facts and hierarchy inference to derive the traffic areas. For the second use case, the reasoning times are longer (above 10 s). Such long reasoning time does not meet the real-time requirements of a smooth traffic flow. This is due to the high number of axioms defined in the ASAM OpenXOntology and hardware constraints. We count more than 2000 axioms that the reasoner has to include to come to the correct conclusion for our query. Possible ways to improve the reasoning speed is to utilize pruning techniques on the core ontology, where large parts of concepts, relations, axioms can be removed or deactivated.

For a complete implementation we should further consider the formalization of the specific lengths of the ratios for broken lines of longitudinal markings. Ratios are implemented in the ontology and define the lane marking type, however, the lengths for these ratios are different in specific situations. The exemplary use case just showcased one situation where the general (urban) guiding line is distinguished with the guiding line to separate bicycle lanes by their ratios.

Another possible step is to formalize the conditions on when certain lane markings appear and when not. Here, we want to mention that guiding lines do not always appear on streets. If the street has only a small width or when the general traffic volume is low, guiding lines are not obligatory. Additionally, roadway boundaries are not needed, if the boundary of the roadway is clearly defined by the traffic infrastructure. An example is the curbside of a sidewalk that clearly designates the roadway boundary. Moreover, it could be formalized that a warning line always precedes a lane boundary, which is interesting information for trajectory planning. Further work should consider formalization of this knowledge into the ontology.

We deem the German Technical Specification as a reliable source of knowledge, however, it must be noted that this document has been created in 1980 and revised in 1993, hence it is old and not up to date in all aspects of lane markings. In 2019, a new document, called *Richtlinien für die Markierung von Straßen Teil A: Autobahnen* (RMS Teil A) [25], was released that amended parts of our document concerning the motorway area. This document is now state-of-the-art, covers the domain of the motorway in more detail, and clearly describes its scope. Further work should consider using this new Technical Specification to update the information on the domain of the motorway. Concerning the Urban and Non-Urban domains, new Technical Specifications are being developed, which will be published in the near future. They will be called *Richtlinien für die Markierung von Straßen Teil L Landstraßen und Teil S Stadt* (RMS Teil L/S). Together, these three Technical Specifications will supersede the current Technical Specification on Lane Markings completely. Hence, future work should focus on formalizing these new documents. Furthermore, the specifications are already more than 40 years old, but there are streets that have been constructed before the first version of these exact specifications has been published. Hence, some streets do not follow these documents and could appear in a different way, however, when streets are restored, depending on the construction body, the Technical Specifications must be followed.

This work presents the concept on formalization of knowledge from the German Technical Specification on Lane Markings. However, the Road and Transportation Research Association (FGSV) publishes many different Technical Specifications on the domain of road traffic. Further work should consider taking these different Technical Specifications of different domains and formalizing this knowledge in corresponding ontologies. There are Technical Specifications that handle further aspects on lane markings like, the Technical Specifications for Securing Work Sites on Roads in Accordance with Traffic Regulations (RSA 21) [26] and Recommendations for Cycling Traffic Infrastructure (ERA) [27]. For Technical Specifications for road signs, we recommend the Technical Specifications for Directional Signage on Motorways (RWBA) [28] and the Technical Specifications for Directional Signage outside Motorways (RWB) [29].

The last discussion point is the scope of countries. These specifications are strictly valid for Germany, hence other countries do not follow these guidelines on lane markings and have their own guidelines. While developing driving automation systems, it must be considered that this knowledge, and therefore the ontology, only applies for Germany and is not useful for other countries. Our suggestion is to create ontologies that are based on this knowledge in a modular way. The core ontology, here the ASAM OpenXOntology, shall remain the same for all countries and for each relevant country a more specific ontology can be created that can easily be integrated into the ASAM OpenXOntology. Corresponding documents must be identified for other countries and formalized as done in this work. If other countries do not have this kind of Technical Specifications, it is advisable to take this work as a template and formalize knowledge corresponding to the German Technical Specifications.

CONCLUSION

In this paper, we show that the formalization of normative knowledge, here the German Technical Specifications on Lane Markings from the Road and Transportation Research Association (FGSV), into an ontology, is a feasible way to support AI models to be rule compliant and more transparent and traceable. The German Technical Specification on Lane Markings is useful, albeit for now a little outdated document that includes information in a structured manner. As the knowledge is structured as hierarchical concepts and relations, complementing with a set of if-then rules, formalization using ontology is an ideal way to represent this knowledge. As the ASAM OpenXOntology is designed as an extensible framework, we can easily integrate our lane markings ontology, based on the German Technical Specifications, into the part of traffic domain ontology. Furthermore, we can extend the ontology with scenario specific concepts and rules to enhance situational awareness. Moreover, we showcased that the ontology can be used to infer information about the traffic area of appearance of lane markings and the type of lane markings using environmental context, despite their long inference time, which can be further improved using ontology pruning techniques.

ACKNOWLEDGEMENT

The research leading to these results is funded by the German Federal Ministry for Economic Affairs and Climate Action within the project "KI Wissen – Entwicklung von Methoden für die Einbindung von Wissen in maschinelles Lernen". The authors would like to thank the consortium for the successful cooperation.

REFERENCES

[1] E. Shi, "User-centered communication of automated driving to promote road safety," in *27th International Technical Conference on the Enhanced Safety of Vehicles (ESV 2023)*, 2023.

[2] A. Kless, "ADAS-Sensordaten im Griff," 2019.

[3] W. J. Yun, M. Shin, S. Jung, S. Kwon, and J. Kim, "Parallelized and randomized adversarial imitation learning for safety-critical self-driving vehicles," *Journal of Communications and Networks*, pp. 1–12, 2022.

[4] "Straßenverkehrs-Ordnung," 2013. https://www.gesetze-im-internet.de/stvo_2013/.

[5] L. Westhofen, C. Neurohr, M. Butz, M. Scholtes, and M. Schuldes, "Using Ontologies for the Formalization and Recognition of Criticality for Automated Driving," *arXiv:2205.01532 [cs]*, May 2022. arXiv: 2205.01532.

[6] M. Scholtes, L. Westhofen, L. R. Turner, K. Lotto, M. Schuldes, H. Weber, N. Wagener, C. Neurohr, M. Bollmann, F. Körtke, J. Hiller, M. Hoss, J. Bock, and L. Eckstein, "6-Layer Model for a Structured Description and Categorization of Urban Traffic and Environment," Feb. 2021. arXiv:2012.06319 [cs].

[7] S. Maierhofer, P. Moosbrugger, and M. Althoff, "Formalization of Intersection Traffic Rules in Temporal Logic," in *2022 IEEE Intelligent Vehicles Symposium (IV)*, pp. 1135–1144, June 2022.

[8] A. J. Bermejo, J. Villadangos, J. J. Astrain, and A. Córdoba, "Ontology Based Road Traffic Management," in *Intelligent Distributed Computing VI* (G. Fortino, C. Badica, M. Malgeri, and R. Unland, eds.), Studies in Computational Intelligence, (Berlin, Heidelberg), pp. 103–108, Springer, 2013.

[9] G. Bagschik, T. Menzel, and M. Maurer, "Ontology based Scene Creation for the Development of Automated Vehicles," in *2018 IEEE Intelligent Vehicles Symposium (IV)*, pp. 1813–1820, IEEE, June 2018.

[10] M. Hulsen, J. M. Zollner, and C. Weiss, "Traffic intersection situation description ontology for advanced driver assistance," in *2011 IEEE Intelligent Vehicles Symposium (IV)*, (Baden-Baden, Germany), pp. 993–999, IEEE, June 2011.

[11] M. Buechel, G. Hinz, F. Ruehl, H. Schroth, C. Gyoeri, and A. Knoll, "Ontology-based traffic scene modeling, traffic regulations dependent situational awareness and decision making for automated vehicles," in *2017 IEEE Intelligent Vehicles Symposium (IV)*, (Los Angeles, CA, USA), pp. 1471–1476, IEEE, June 2017.

[12] *Richtlinien für die Markierung von Straßen Teil 1: Abmessungen und geometrische Anordnung von Markierungszeichen*. Forschungsgesellschaft für Straßen- und Verkehrswesen, FGSV Verlag, 1993.

[13] S. Staab and R. Studer, eds., *Handbook on Ontologies*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009.

[14] T. R. Gruber, "A Translation Approach to Portable Ontology Specifications," *Knowledge Acquisition*, vol. 5, no. 2, pp. 199–220, 1993.

[15] A. Armand, D. Filliat, and J. Ibanez-Guzman, "Ontology-based context awareness for driving assistance systems," in *2014 IEEE Intelligent Vehicles Symposium Proceedings*, (MI, USA), pp. 227–233, IEEE, June 2014.

[16] G. D. Giacomo and M. Lenzerini, "TBox and ABox Reasoning in Expressive Description Logics," p. 12.

[17] H. Stuckenschmidt, "Debugging OWL Ontologies - A Reality Check," Jan. 2008.

[18] "Protégé Project Website." https://protege.stanford.edu/.

[19] M. A. Musen, “The protégé project: a look back and a look forward,” *AI Matters*, vol. 1, no. 4, pp. 4–12, 2015.

[20] “Document FRAV-02-05/Rev.2,” 2020. https://wiki.unece.org/display/trans/FRAV+2nd+Session.

[21] Y. Wang, M. Grabowski, H. Su, and A. Paschke, “An Ontology-based Model for Handling Rule Exceptions in Traffic Scenes,” in *International Workshop on AI compliance mechanism (WAICOM 2022)*, 2022.

[22] “SWRL: A Semantic Web Rule Language Combining OWL and RuleML.” https://www.w3.org/Submission/SWRL/.

[23] “SPARQL: Query Language for RDF.” https://www.w3.org/TR/rdf-sparql-query/.

[24] S. Union, “The Enterprise Knowledge Graph Platform | Stardog.” https://www.stardog.com/.

[25] *Richtlinien für die Markierung von Straßen (RMS) - Teil A: Autobahnen*. Forschungsgesellschaft für Straßen- und Verkehrswesen, FGSV Verlag, 2019.

[26] *Richtlinien für die verkehrsrechtliche Sicherung von Arbeitsstellen an Straßen*. Forschungsgesellschaft für Straßen- und Verkehrswesen, FGSV Verlag, 2021.

[27] *Empfehlungen für Radverkehrsanlagen*. Forschungsgesellschaft für Straßen- und Verkehrswesen, FGSV Verlag, 2010.

[28] *Richtlinien für die wegweisende Beschilderung auf Autobahnen*. Forschungsgesellschaft für Straßen- und Verkehrswesen, FGSV Verlag, 2000.

[29] *Richtlinien für die wegweisende Beschilderung außerhalb von Autobahnen*. Forschungsgesellschaft für Straßen- und Verkehrswesen, FGSV Verlag, 2000.

SAFE CONTROL TRANSITIONS: MACHINE VISION BASED OBSERVABLE READINESS INDEX AND DATA-DRIVEN TAKEOVER TIME PREDICTION

Ross Greer
Nachiket Deo
Akshay Rangesh
Mohan Trivedi
Laboratory for Intelligent & Safe Automobiles[1]
University of California San Diego
USA

Pujitha Gunaratne
Toyota Collaborative Safety Research Center
USA

Paper Number 23-0331

ABSTRACT

To make safe transitions from autonomous to manual control, a vehicle must have a representation of the awareness of driver state; two metrics which quantify this state are the Observable Readiness Index and Takeover Time. In this work, we show that machine learning models which predict these two metrics are robust to multiple camera views, expanding from the limited view angles in prior research. Importantly, these models take as input feature vectors corresponding to hand location and activity as well as gaze location, and we explore the tradeoffs of different views in generating these feature vectors. Further, we introduce two metrics to evaluate the quality of control transitions following the takeover event (the maximal lateral deviation and velocity deviation) and compute correlations of these post-takeover metrics to the pre-takeover predictive metrics.

INTRODUCTION

It is important to plan for safe operation of intelligent vehicles in situations of system failure. Intelligent and autonomous vehicles face challenges when dealing with long-tail events, defined as events which occur with little to no regularity and are thus difficult for the dominant regime of learning-based perception and control models to operate safely. When such situations are identified, the vehicle may benefit from passing control to the human driver. However, there is risk in such control transitions, especially if the driver is not alert to the scene or ready to operate the vehicle; in these cases, it may be safer for the vehicle to perform an emergency maneuver such as braking or pulling over.

To mediate between these options, predicting the driver’s takeover readiness is a critical human factor consideration for safe control transitions in conditionally autonomous vehicles. The duration of such a transition is quantified by the so-called Takeover Time (TOT), which measures the interval between an autonomous vehicle’s request for manual driving (Takeover Request or TOR) and the time the driver assumes control. An illustration of the role of Takeover Time when an autonomous vehicle encounters an on-road hazard (and the necessary computation of the driver’s ability to safely perform that takeover) are provided in Figure 1.

[1] cvrr.ucsd.edu

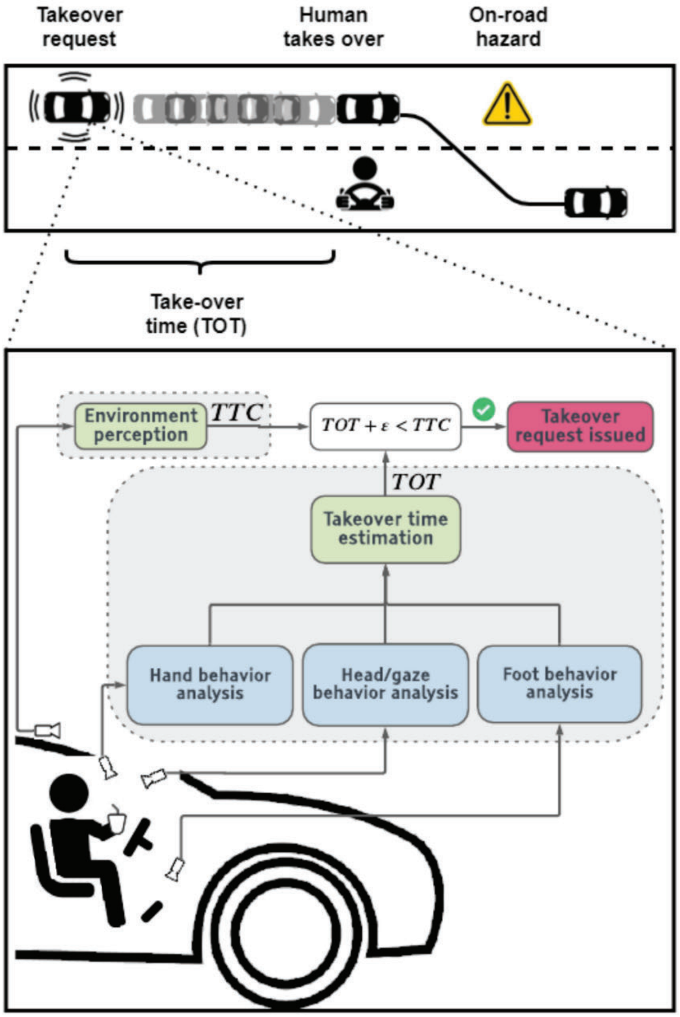


Figure 1. An autonomous vehicle may issue a takeover request when encountering an on-road hazard. For a safe control transition, the vehicle must have an understanding of the driver's state, which can be inferred via in-cabin cameras observing the driver for visual cues.

Establishing metrics to evaluate the effectiveness of a control transition is critical for effective experimentation and design of the systems we trust to make these decisions safely. Ego-vehicle metrics based on lateral and longitudinal motion immediately post-takeover provide information on how smoothly and safely the driver has established manual control.

Altogether, the problem of predicting readiness for control transition requires metrics across a before-during-after framework: how ready is the driver to take control (before), how long will it take the driver to establish control (during), and how successfully did the driver take control (after).

RELATED RESEARCH

Driving in real freeway and urban environments can subject drivers (and autonomous agents) to a variety of complex scenes, such as unexpected surround vehicle path changes and marked and unmarked intersections. Many research works seek to better analyze and understand the surrounding scene to make for safer autonomous interactions [1, 2, 3, 4], but of equal importance is the ability of the vehicle to provide transition of control to an alert driver to complete interactions under uncertainty. Driver state monitoring [5, 6] has many safety applications in crash prevention and mitigation under manual control, but here we focus on situations where vehicles begin in an autonomous state and return control to a human driver for safety. Commercial trends show growth in the use of driver-facing camera systems to facilitate this state monitoring, as both safety benefits of such systems and privacy-preserving mechanisms [7] improve. However, a recent study [8] determined that cars with such systems exhibited both inconsistent and unsafe behaviors as well as poor driver alerting on road departure and construction zone test contexts for a highly-automated vehicle. Further research shows that even when a takeover is enacted, there is significant risk of accidents even after driver intervention [9], though research has guided effective HMI to communicate the vehicle state (and request for manual control) to the driver [10, 11, 12]. Efforts towards improving safety in takeover situations align with the Fallback (Minimal Risk Condition) category in the NHTSA framework for Automated Driving Systems [13], further exemplified in the ADS test framework proposed by Thorn et al. [14].

Deo and Trivedi [15] define the Objective Readiness Index (ORI), a metric which quantifies a driver's readiness behind the wheel by normalizing and averaging ratings assigned by multiple human observers viewing feeds from in-cabin cameras capturing the driver's gaze, hand, and foot activity. Research connecting these cues to driver attention have provided an important basis for driver state analysis in safety applications [16, 17]. They show this metric can be predicted using an LSTM-based machine learning model inferring on observations from similar in-cabin camera feeds. Recent successful predictive methods encode such in-cabin camera observations into class probability vectors using convolutional neural networks. These include Rangesh and Trivedi's *HandyNet* CNN [18], Yuen and Trivedi's part affinity fields approach [19], Vora et al.'s gaze CNN [20], and Rangesh and Trivedi's forced spatial attention approach [21]. Such vectors serve as low-dimensional representations of the predictive information from appearance-based models of hand, eye, and foot activity of the driver.

Extending to an additional objective metric, Rangesh et al. [22, 23] show that computer vision and machine learning algorithms can be used to predict quantities such as the takeover time for a driver during a transition from autonomous to manual control. Both ORI and TOT estimation methods use as features the estimated positions of the driver's eye gaze, hands, and feet relative to the driving scene, steering wheel, and pedals respectively.

While many studies analyze driver takeovers from simulation, our research is conducted over naturalistic driving data collected from real autonomous vehicles operating on an experimental test track. Shi and Bengler [24] provide analysis of takeover times in relationship to external conditions and in-cabin driver tasks; we provide similar analysis and include learning-based methods of estimating a driver's readiness and takeover time.

Previous works demonstrate the ability of such algorithms to operate within a fixed experimental camera view; here we explore whether algorithms which predict takeover readiness generalize well to multiple driver views. Further, we explore the relationship between takeover readiness and the timing and *quality* of such takeover events. Takeover quality is of critical importance, as there is apparent difference in the reflexive establishment of motor readiness versus cognitive processing of a road situation which can be impaired by driver distraction [25].

METHODS

While related works collect gaze and hand features from a single camera view, here we adopt a multi-camera framework for evaluation. We apply convolutional models to estimate driver gaze and hand states from a variety of camera views, then further illustrate the effectiveness of these model outputs in estimating ORI and TOT, following the machine learning framework defined in [15, 22, 23].

Models For Driver Behavior Analysis

To analyze driver behavior in estimating ORI, we first collect frame-wise output features of driver gaze and hand activity, using the pipeline depicted in Figure 2. The method we employ begins with detection of the driver in each of the four camera views, followed by application of a human-pose estimation model to predict the driver's joint locations. Such an approach is traditionally referred to as "Top-Down". It is worth noting that these joint detection models are sensitive to the accuracy of the initial detection of the driver, and likewise the models for driver activity classes are sensitive to the accuracy of the detected joints.

For driver detection, we employ the MMDetection [26] implementation of Faster-RCNN [27] with Feature Pyramid Networks [28], using a ResNet-50 backbone [29] pretrained on COCO [30]. For joint detection, we employ the MMPose [31] implementation of HRNet [32], also pretrained on COCO human detections fit to a resolution of 256x192. We use the driver's ears, eyes and nose key-points to localize and crop their eye region. The cropped image is then passed through a convolutional neural network that classifies the driver's gaze zone. Similarly, we use the driver's elbow and wrist key-points to localize their hands. Cropped images of the driver's hands are passed through convolutional neural networks that classify the driver's hand location (e.g. on wheel, interacting with infotainment, in-air gesturing, etc.) and held objects (e.g. phone, beverage, etc.). Figure 3 shows the driver's eye and hands localized using their pose keypoints.

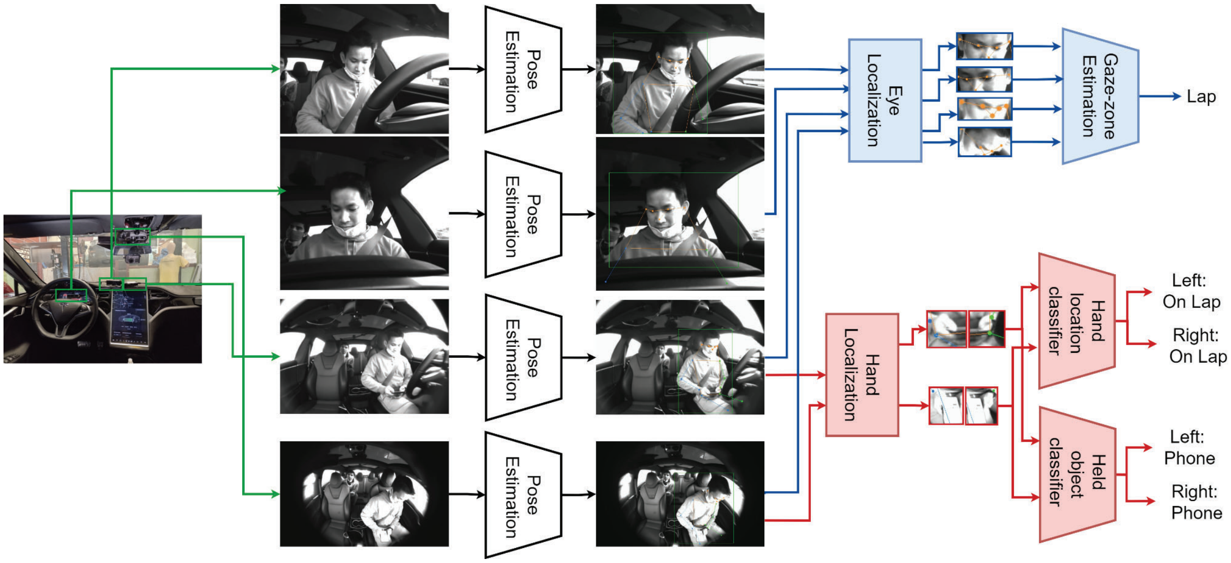


Figure 2. Overview of driver behavior analysis: We apply a detector and human-pose estimation model to each of the four views to localize the driver and their joints. We use the pose key-points to localize the driver's eyes and hands. Cropped images around the driver's eyes and hands are passed through convolutional neural networks to classify the driver's gaze zone, hand location, and held object.

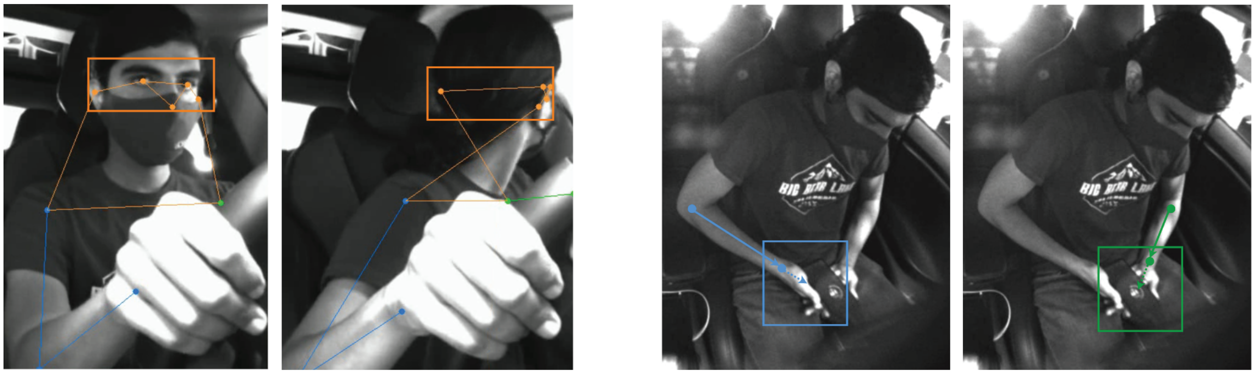


Figure 3. Eye and hand localization: We localize the drivers eyes and hands using the estimated head, elbow, and wrist key-points.

Dataset We collected gaze data from 9 different subjects for training the gaze and hand zone estimation models. Each subject was instructed to look at the different gaze zones sequentially by an experimenter. The subjects were encouraged to vary their head and body pose while ensuring that their gaze was directed towards the same gaze zone. Likewise, subjects were asked to place their hands through five hand locations: on steering wheel, on lap, in air (including gesturing), using infotainment unit, and on cupholder, followed by three held object activities: phone, tablet, and beverage (water bottle). The captured videos were then split into contiguous segments for each gaze zone, hand zone, and held object, providing labeled training data for the models. All 4 cameras synchronously captured the training data, yielding a total of 86953 frames with labeled gaze zones, 181,584 frames with labeled hand zones, and 263,166 images per camera with labeled held objects. During training for the gaze and hand classification systems, we employed a cross-validation method whereby one subject is entirely removed from the training data, and used only as the evaluation subject. This technique helps in identifying the generalization of the models to subjects unseen in training data (i.e. avoiding overfitting to particular participants' hands and eyes).

Gaze Zone Classification We use the cropped image around the driver's eyes to classify the driver's gaze direction into one of five different gaze zones. We consider the following gaze zones related to driving as well as non-driving activities: forward, rearview (generalized to include left shoulder/blind spot, left mirror, rearview mirror, right mirror, and right shoulder/blind spot), lap, speedometer, and infotainment. Figure 4 shows an illustration of the driver looking at each of the 5 gaze zones in all 4 views. These set of gaze zones capture non-driving related tasks (NDRTs) such as interacting with the infotainment unit or using a handheld device, as well as an attentive driver checking the vehicle's surroundings in the front or rear.

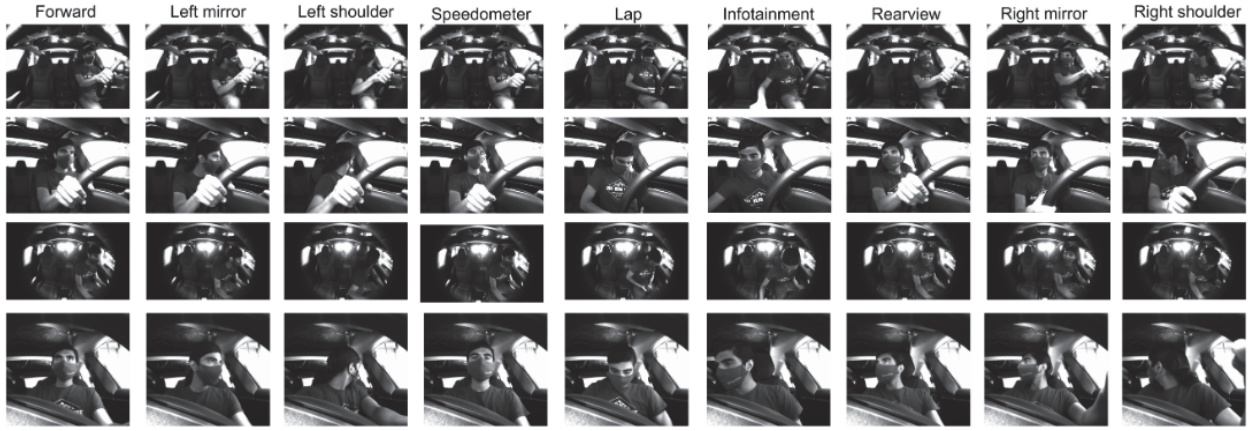


***Figure 4.* The 5 gaze zones as seen from our 4 camera views.**

We used an EfficientNet-B3 [33] model for classifying gaze zones. We used the ImageNet pretrained weights as the starting point for training and trained the model using our collected training data. The final fully connected layer was modified to have 5 outputs for the 5 gaze zones. We trained separate models for each of the four camera views: dashboard-center, dashboard-driver, steering, and rearview. Table 1 shows the classification accuracies with each camera view. Figure 5 shows the confusion matrices for gaze zone classification with the 4 camera views. From the confusion matrices we note that the dashboard driver and steering camera views achieve high accuracies for all gaze zones, compared to the rearview and dashboard-center camera views. Further, the most commonly confused gaze zones are (i) speedometer and forward and (ii) rearview mirror and infotainment. Both of these views correspond to slight differences in head/gaze pitch. The steering column camera which is directly pointed at the driver's face (see Fig. 4) is best suited for distinguishing between these gaze zones.

Table 1
Gaze Zone Classification Accuracies (%)

View	EfficientNet-B3 5-class
Dashboard-driver	89.71
Dashboard-center	85.31
Rearview	83.09
Steering	**91.41**

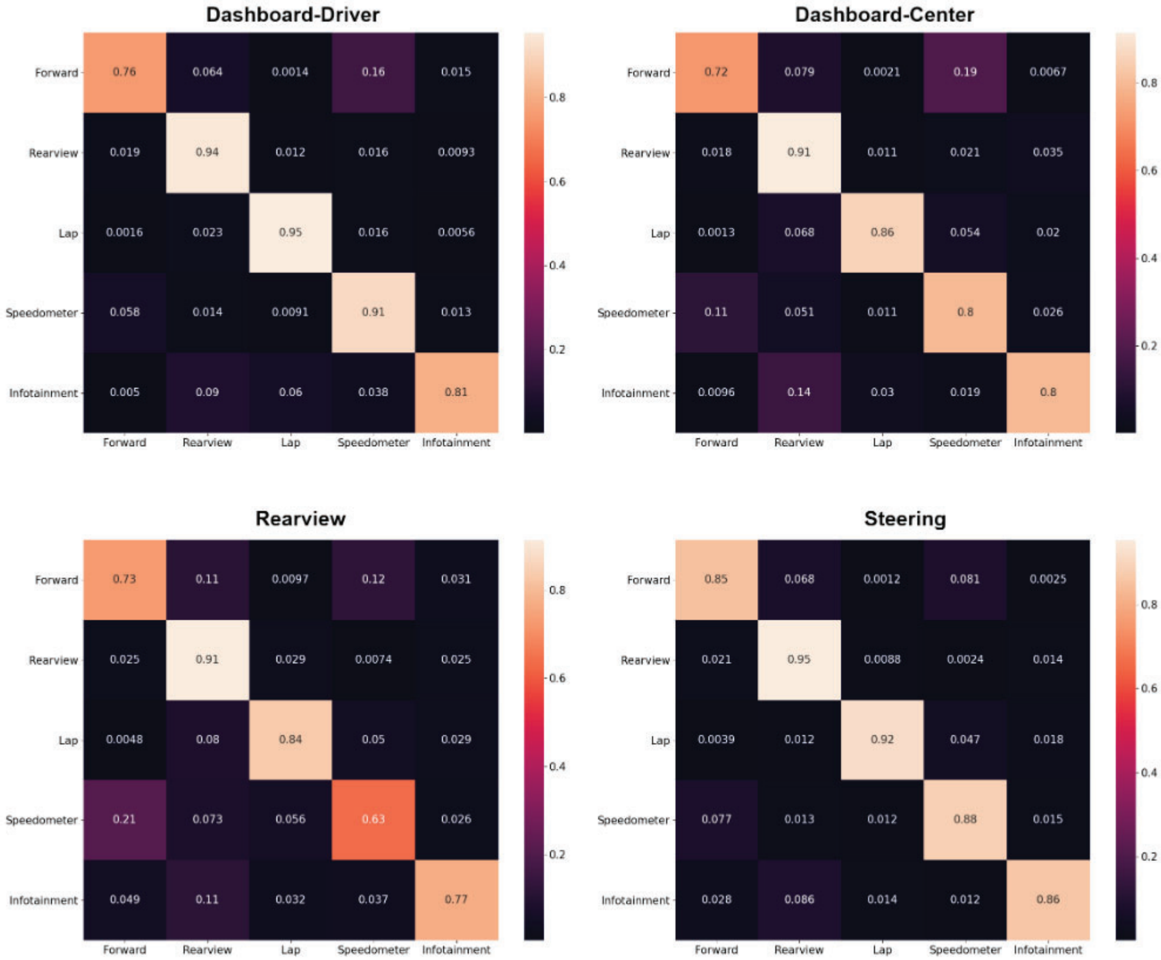


Figure 5. Confusion matrices for gaze zone classification using the 4 different camera views.

<u>Hand Location and Held Object Classification</u> We use cropped images around the driver's hands to classify the driver's hand locations and held objects using a ResNet-18 model (pretrained on ImageNet). We use a separate model per hand and per function, yielding a four model suite per camera view. Additionally, for each camera view, we use three different crop sizes, as different views may benefit from larger or more constrained contextual regions, and pixel area occupied by the hand may change between views. We sampled from dimensions 50x50, 100x100, and 200x200, and recommend that these hyperparameters can be tuned specific to the camera view and desired task.

With the aforementioned group of combinations, we train a total yield of 48 models. Of these, we select the best-performing model across the crop sizes, with the results reported in the confusion matrices shown in Figures 6 and 7. From these matrices, the dashboard center view provides the best estimate of left and right hand positions, while the rearview and dashboard center cameras appear to perform better for held object classification. For the driver-center view, left hand zone classification accuracy is 79% and right hand zone classification accuracy is 90% corresponding to the confusion matrices in Figure 6. In the rearview and dashboard center cameras respectively, left held object classification accuracy is 75% and right held object classification accuracy is 72% corresponding to the confusion matrices in Figure 7. One promising feature of the held-object classifiers is that the models provide a fairly consistent binary function between whether an object is held or not held (that is, it tends to correctly return "None" when no object is held, and some object when any object is held).

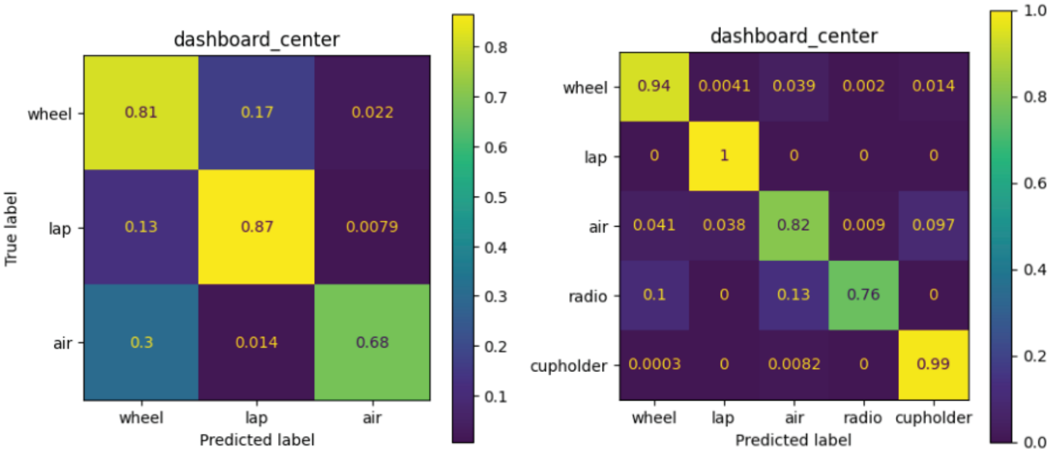


Figure 6. Confusion matrices for left hand zone classification (left) and right hand zone classification (right) from the experimentally-optimal dashboard center view. Note that the left hand is excluded from reaching the radio or cupholder in our current scheme.

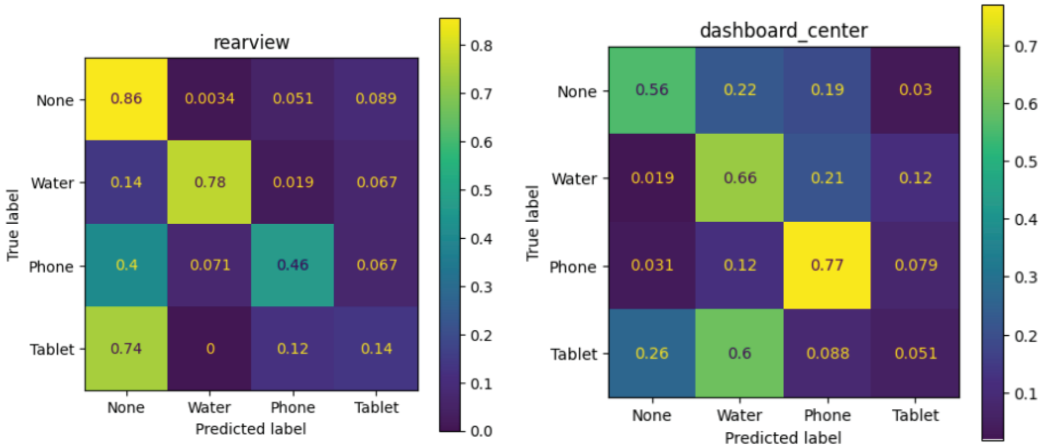


Figure 7. Confusion matrices for left hand held object classification (left) and right hand object classification (right) from the experimentally optimal camera views.

EXPERIMENTS AND EVALUATION

Observable Readiness Index Estimation with Multi-Camera Framework for In-Cabin Activity Monitoring

The Observable Readiness Index (ORI) estimation model consists of two steps, shown in the red blocks of Figure 8. The first step involves extracting frame-wise features capturing the driver's state, and the second step involves using an LSTM model to aggregate temporal context over the past 2 seconds of frame-wise features. The original ORI model described in [15] used frame-wise features from 4 cameras observing the driver's face, hands, feet and body pose, and two IR range sensors observing the driver's hands and feet. To adapt it to the above multi-camera framework, we train the ORI model using a subset of features focused on driver gaze and hand activity, obtained from models described in the above sections for gaze features (gaze zone probabilities) and hand features (held object class probabilities and hand activity class probabilities).

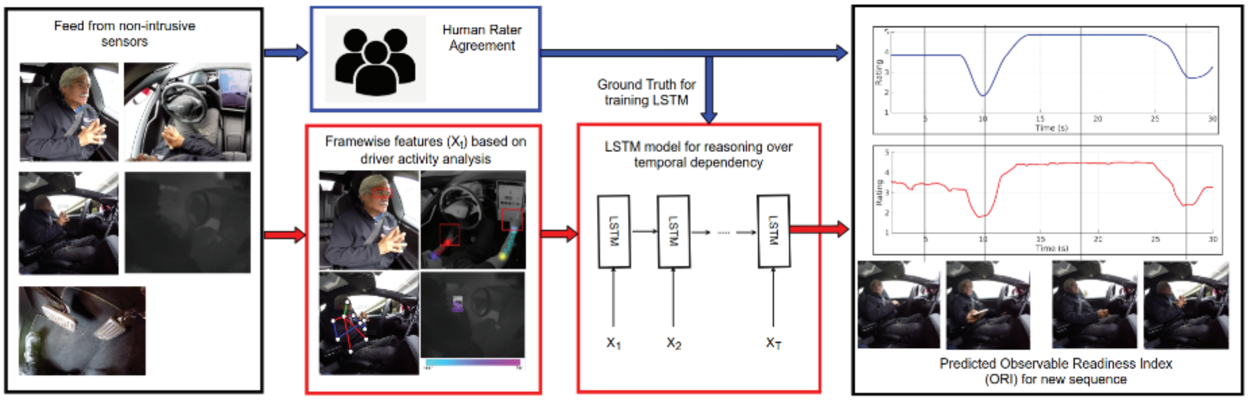

Feed from non-intrusive sensors
Human Rater Agreement
Ground Truth for training LSTM
Framewise features (X_t) based on driver activity analysis
LSTM model for reasoning over temporal dependency
LSTM
LSTM
LSTM
X_1
X_2
X_T
Rating
Time (s)
Rating
Time (s)
Predicted Observable Readiness Index (ORI) for new sequence

Figure 8. Model flow for ORI Estimation.

Datasets

Naturalistic driving data for training vision models was collected from two source testbeds: (1) the LISA-T testbed [17] operated in urban and freeway environments in La Jolla, California (USA), and (2) the controlled driving datasets collected from a twin Tesla Model S described in [15, 22, 23], capturing real takeover events from over 100 subjects on a test track in Iowa. Both testbeds have self-driving features which can be operated in real-world freeway environments. Subjects in Iowa performed each of eight different non-driving related tasks (NDRTs) while the Tesla operated on autopilot, maintaining its lane and a cruising speed of 30 mph. Subjects were then issued a TOR while the autopilot was simultaneously disengaged. The drivers are expected to assume control and stabilize the vehicle. Data from the Iowa test track further included distances to lane markings, captured from an onboard Mobileye system, and speed data annotated from the dashboard speedometer.

The LISA-T testbed dataset, used here to evaluate the effectiveness of the multi-camera framework and relationship of readiness to takeover quality, features collected video data on three drivers from four infrared cameras, mounted behind the steering wheel, on the dashboard facing the driver, on the dashboard facing the center of the cabin, and from the rearview mirror facing the center of the cabin (the same positions used in training the gaze and hand models described in the previous section). This naturalistic driving dataset consists of roughly 10 hours of driving data with LISA-T operating in autonomous mode in slow-moving traffic on freeways, with three different drivers. The drivers perform non-driving related tasks (NDRTs) such as operating the infotainment unit to navigate to a specific location, changing the radio station, using a hand-held device to read a text and drinking from cup/bottle. A safety co-passenger constantly monitors the road and vehicle state when the driver performs the NDRT. To simulate takeover requests, the safety co-passenger triggers a TOR, after which the driver is instructed to bring their eyes on the road and hands on the wheel. Note that control is not transferred from the vehicle to the driver after takeover alarms (as in the controlled driving dataset on the Iowa test track) since this would be unsafe in real traffic. Our dataset contains 295 total "takeovers". We extract 30 second clips around each "takeover" event. Two second snippets from these clips are rated by raters on a scale of 1 to 5, as illustrated in Figure 9. The ratings and normalized and interpolated over the video clips to obtain the ground truth ORI ratings.

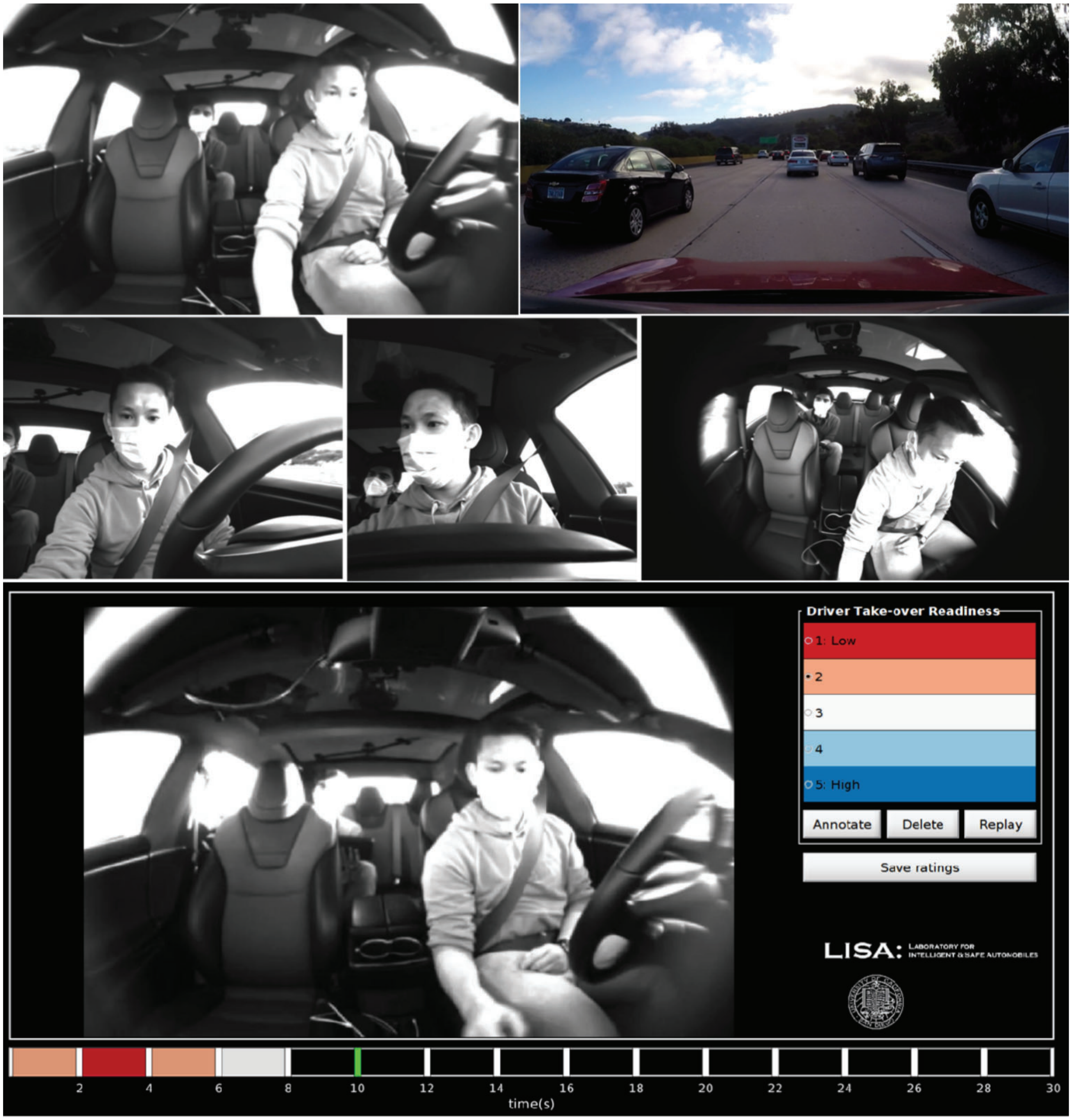


Figure 9. Raters provide a rating on a scale of 1 to 5 to describe the readiness of the driver over each two second interval of a 30 second driving clip, as described in [1].

Observable Readiness Index Estimation

We train the ORI estimation model using a subset of features (gaze zones, held-objects and hand activity) using naturalistic data from LISA-T and the controlled driving datasets. Note that the extracted features, namely, gaze zone probabilities, held-object probabilities and hand activity probabilities, are all view independent, allowing for training on data from testbeds with differing camera configurations.

We compare three different variants of the ORI model, which use purely gaze features, purely hand features, and both sets of features. Additionally, we compare the camera views used for extracting gaze features and hand features. Table 2 reports the mean absolute errors between the ground truth ORI ratings and the predicted ORI values for the newly collected dataset with the multi-camera framework.

Table 2

Mean average error for ORI estimation under different combinations of features and views

Gaze Features				Hand features		
Dashboard Driver	Dashboard Center	Rearview	Steering	Dashboard Center	Rearview	MAE
✓						**0.6711**
	✓					0.7670
		✓				0.7404
			✓			0.6883
				✓		1.5171
					✓	1.5722
	✓			✓		0.9215
		✓			✓	0.9280

Experimental results suggest the ORI models that purely use gaze features outperform those that purely use hand features, as well as those that use a combination of both sets of features. In other words, the hand features seem to be adversely affecting ORI estimation. However, previous ORI experiments [15] clearly show the utility of hand locations and held objects for estimating ORI. The poor results with hand features suggest two possibilities: first, that the hand location and held object classifier modules may need to be further trained to improve their accuracies, and second, that the particular selected camera views may not be the most suitable for driver hand analysis, because the driver's hands are only visible in the dashboard-center and rearview cameras (in which views the driver's hands are often occluded or truncated). Within the purely gaze-based ORI estimation models, the models that use the dashboard-mounted driver-facing camera and steering wheel view achieve the lowest MAE values, as expected from views which have the most direct visibility of the eyes.

Objective Takeover Readiness Metrics Using Ego-Vehicle State

While the ORI represents prediction of a quantity derived from subjectivity, here we introduce an objective measure of takeover quality. A safe takeover performed by a prepared driver would appear seamless, meaning that the ego vehicle would move in a predictable manner, without sudden braking, acceleration or lateral deviation. A distracted driver, on the other hand, may overreact, leading to unpredictable longitudinal or lateral motion of the ego vehicle. The ego-vehicle's motion during takeovers can thus provide a useful objective measure of driver takeover readiness.

We derive two objective readiness measures from the ego-vehicle state: one corresponding to longitudinal motion, and one corresponding to lateral motion. First, we consider the maximum change in speed (Δv) of the vehicle during the immediate 5 seconds post TOR, intended to capture any sudden braking or acceleration by an under-prepared driver. Second, we consider maximum deviation from the lane centerline (Δx) during the immediate 5 seconds post TOR, intended to capture any sudden swerving or drift during the takeover.

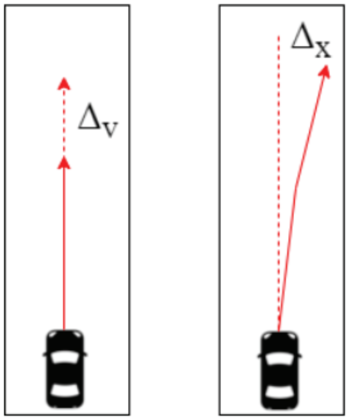


Figure 10. As objective metrics based on ego-vehicle state, we measure the maximum deviation in ego vehicle speed (Δv) 5 seconds immediately after the TOR (left), and maximum lateral deviation from lane centerline (Δx) 5 seconds immediately after the TOR (right).

From the collected data, we measure the maximum change in speed, and deviation from lane centerline 5 seconds post TOR for each experimental trial (Δv and Δx). Figures 11 and 12 show the average values of Δv and Δx by NDRT respectively.

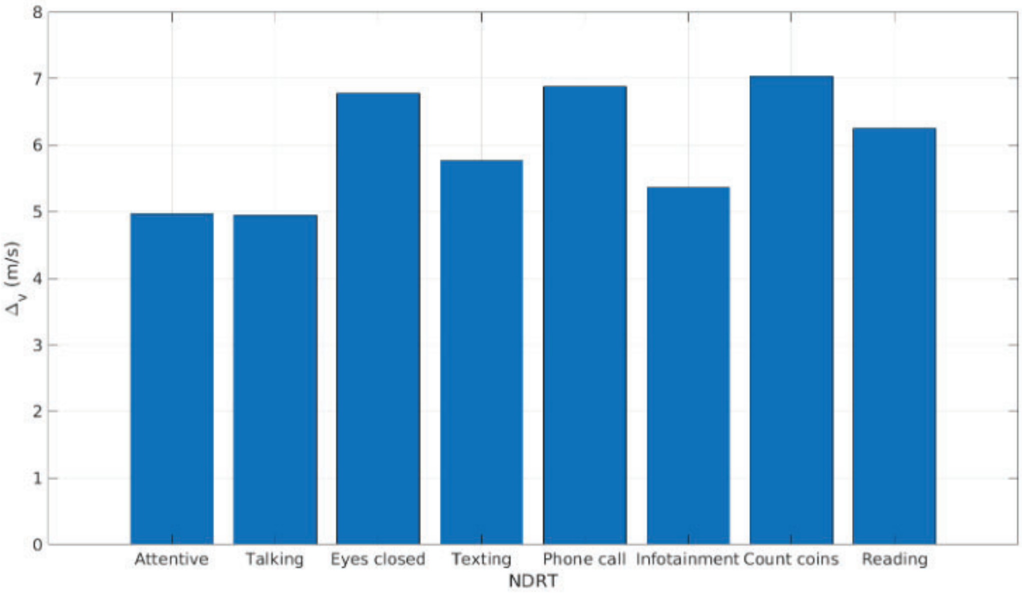


Figure 11. Max deviation in speed up to 5 seconds post TOR (Δv) by NDRT.

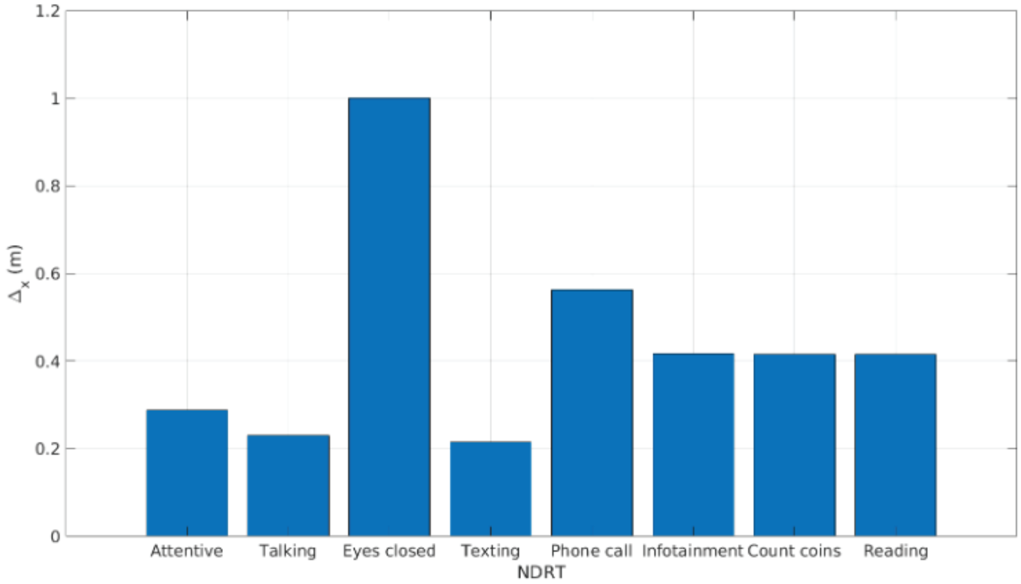


Figure 12. Max deviation from lane centerline up to 5 seconds post TOR (∆x) by NDRT.

We note that ∆v is lowest when the driver is attentive or just talking to co-passenger, while ∆v is high for highly distracting NDRTs such as eyes closed, phone call, reading, counting coins and texting. This suggests that the driver tends to overreact post takeover when distracted. A similar trend can be observed for ∆x, with the maximum lateral deviation observed for the eyes closed (sleeping) NDRT.

Correlation of Objective Readiness Metrics with ORI and Takeover Time

We compute the correlation of ORI prior to TOR with the objective metrics ∆v and ∆x, as well as the correlation of ∆v and ∆x with takeover time, reported in Table 3.

Table 3

Correlation of ∆x and ∆x with ORI and Takeover Time

	Δ_v	Δ_x
ORI	-0.0467	-0.0427
Takeover time	0.1686	0.0072

We note that we get very slight negative correlations for ORI with ∆v and ∆x, meaning a low value of observable readiness corresponds to high deviations in speed and high deviations from the lane centerline. On the other hand, takeover times have a slight positive correlation with ∆v, meaning high takeover times correspond to high values of ∆v . From our experiments, ∆x is uncorrelated with takeover time.

CONCLUDING REMARKS

This exploration suggests that the ORI model generalizes to various views of the cabin and serves as an effective predictor of TOT, but relies on the performance of the modules for feature extraction from the hands and eyes. Further investigation in methods for improved accuracy of hand zone and held object classifiers would serve to further improve downstream models' (such as ORI) predictive abilities. These experiments suggest that there is a tradeoff in the ability of a camera to accurately observe both the eyes and the hands, and that careful consideration should be made when selecting camera positions intended to predict driver readiness, and that the optimal placement is highly task-dependent.

In analyzing takeover quality as a function of readiness and estimated takeover time, we find the trends in correlation to match intuition; the negative correlations between ORI and takeover quality metrics show that low readiness corresponds to high deviations in vehicle lateral position and longitudinal velocity, and positive correlations between TOT and takeover quality metrics show that quick takeovers correspond to less deviations in vehicle lateral position and longitudinal velocity.

We note the low absolute values of all correlation coefficients, suggesting that the link between the objective readiness metrics and ORI/takeover time is inconclusive. One prominent source of noise is that some drivers brake and bring the ego-vehicle to a halt post TOR in the controlled driving dataset, rather than maintaining ego-vehicle speed. A more precise experiment where the driver is instructed to maintain the ego-vehicle's motion after the TOR may yield more reliable metrics of takeover quality. Additionally, surrounding traffic, which is missing in the controlled driving environment, may affect these values, as would the next navigational goal of the ego vehicle. An initial study varying these parameters in a simulator setting followed by a real world study with an appropriate safety protocol (e.g. secondary driver with pedal controls) could yield results of further interest.

In conclusion, the experimental analysis represents a complete and fully-AI-driven instance of the before-during-after analytical framework for safer control transitions in real-world takeover events; predicted ORI speaks to readiness "before" a takeover, predicted TOT speaks to pace of transition "during" the takeover, and ego vehicle motion metrics speak to the quality of control "after" the takeover.

REFERENCES

[1] Møgelmose, A., Trivedi, M. M., & Moeslund, T. B. (2015, June). Trajectory analysis and prediction for improved pedestrian safety: Integrated framework and evaluations. In 2015 IEEE intelligent vehicles symposium (IV) (pp. 330-335). IEEE.

[2] Greer, R., & Trivedi, M. (2022). From Pedestrian Detection to Crosswalk Estimation: An EM Algorithm and Analysis on Diverse Datasets. arXiv preprint arXiv:2205.12579.

[3] Trivedi, M. M., Gandhi, T. L., & Huang, K. S. (2005). Distributed interactive video arrays for event capture and enhanced situational awareness. IEEE Intelligent Systems, 20(5), 58-66.

[4] Greer, R., Deo, N., & Trivedi, M. (2021). Trajectory prediction in autonomous driving with a lane heading auxiliary loss. IEEE Robotics and Automation Letters, 6(3), 4907-4914.

[5] Ohn-Bar, E., Tawari, A., Martin, S., & Trivedi, M. M. (2015). On surveillance for safety critical events: In-vehicle video networks for predictive driver assistance systems. Computer Vision and Image Understanding, 134, 130-140.

[6] Cheng, S. Y., Park, S., & Trivedi, M. M. (2007). Multiperspective and multimodal video arrays for 3D body tracking and activity analysis. Comput. Vis. Image Underst.(Special Issue on Advances in Vision Algorithms and Systems Beyond the Visible Spectrum), 106(2-3), 245-257.

[7] Martin, S., Tawari, A., & Trivedi, M. M. (2014). Toward privacy-protecting safety systems for naturalistic driving videos. IEEE Transactions on Intelligent Transportation Systems, 15(4), 1811-1822.

[8] Cummings, M. L., & Bauchwitz, B. (2021). Safety Implications of Variability in Autonomous Driving Assist Alerting. IEEE Transactions on Intelligent Transportation Systems.

[9] Karakaya, B., & Bengler, K. (2021, June). Investigation of driver behavior during minimal risk maneuvers of automated vehicles. In Congress of the International Ergonomics Association (pp. 691-700). Springer, Cham.

[10] Naujoks, F., Hergeth, S., Keinath, A., Wiedemann, K., & Schömig, N. (2019). Development and Application of an Expert Assessment Method for Evaluating the Usability of SAE Level 3 Ads HMIs. System, 3, L2.

[11] Daman, P., Götze, M., Gold, C., & Kompass, K. (2019). BMW's Safety Guidelines for the Testing and Deployment of Automated Vehicles. In 26th International Technical Conference on the Enhanced Safety of Vehicles (ESV): Technology: Enabling a Safer TomorrowNational Highway Traffic Safety Administration (No. 19-0226).

[12] Kurpiers, C., Lechner, D., & Raisch, F. (2019, June). The influence of a gaze direction based attention request to maintain mode awareness. In Proceedings of the 26th International Technical Conference on the Enhanced Safety of Vehicles, Eindhoven, The Netherlands (pp. 10-13).

[13] National Highway Traffic Safety Administration. (2017). Automated driving systems 2.0: A vision for safety. Washington, DC: US Department of Transportation, DOT HS, 812, 442.

[14] Thorn, E., Kimmel, S. C., Chaka, M., & Hamilton, B. A. (2018). A framework for automated driving system testable cases and scenarios (No. DOT HS 812 623). United States. Department of Transportation. National Highway Traffic Safety Administration.

[15] Deo, N., Trivedi, M.M.: Looking at the driver/rider in autonomous vehicles to predict take-over readiness. IEEE Transactions on Intelligent Vehicles 5(1), 41–52 (2019)

[16] Victor, T., Dozza, M., Bärgman, J., Boda, C. N., Engström, J., Flannagan, C., ... & Markkula, G. (2015). Analysis of naturalistic driving study data: Safer glances, driver inattention, and crash risk (No. SHRP 2 Report S2-S08A-RW-1).

[17] Rangesh, A., Deo, N., Yuen, K., Pirozhenko, K., Gunaratne, P., Toyoda, H., & Trivedi, M. M. (2018, November). Exploring the situational awareness of humans inside autonomous vehicles. In 2018 21st International Conference on Intelligent Transportation Systems (ITSC) (pp. 190-197). IEEE.

[18] Rangesh, A., Trivedi, M.M.: Handynet: A one-stop solution to detect, segment, localize & analyze driver hands. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. pp. 1103–1110 (2018)

[19] Yuen, K., Trivedi, M.M.: Looking at hands in autonomous vehicles: A convnet approach using part affinity fields. IEEE Transactions on Intelligent Vehicles 5(3), 361–371 (2019)

[20] Vora, S., Rangesh, A., and Trivedi, M.M.: "Driver Gaze Zone Estimation using Convolutional Neural Networks: A General Framework and Ablative Analysis," IEEE Transactions on Intelligent Vehicles, 2018.

[21] Rangesh, A., and Trivedi, M.M.: "Forced Spatial Attention for Driver Foot Activity Classification," ICCV Workshop on Assistive Computer Vision and Robotics (Oral), 2019.

[22] Rangesh, A., Deo, N., Greer, R., Gunaratne, P., Trivedi, M.M.: Autonomous vehicles that alert humans to take-over controls: Modeling with real-world data. In: 2021 IEEE International Intelligent Transportation Systems Conference (ITSC). pp. 231–236. IEEE (2021)

[23] Rangesh, A., Deo, N., Greer, R., Gunaratne, P., Trivedi, M.M.: Predicting take-over time for autonomous driving with real-world data: Robust data augmentation, models, and evaluation. arXiv preprint arXiv:2107.12932 (2021)

[24] Shi, E., & Bengler, K. (2022). Non-driving related tasks' effects on takeover and manual driving behavior in a real driving setting: A differentiation approach based on task switching and modality shifting. Accident Analysis & Prevention, 178, 106844.

[25] Zeeb, K., Buchner, A., & Schrauf, M. (2016). Is take-over time all that matters? The impact of visual-cognitive load on driver take-over quality after conditionally automated driving. Accident analysis & prevention, 92, 230-239.

[26] Chen, K., Wang, J., Pang, J., Cao, Y., Xiong, Y., Li, X., Sun, S., Feng, W., Liu, Z., Xu, J., Zhang, Z., Cheng, D., Zhu, C., Cheng, T., Zhao, Q., Li, B., Lu, X., Zhu, R., Wu, Y., Dai, J., Wang, J., Shi, J., Ouyang, W., Loy, C., and Lin, D.: Open MMLab Detection Toolbox and Benchmark. arXiv preprint arXiv:1906.07155 (2019)

[27] Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. Advances in neural information processing systems, 28.

[28] Lin, T. Y., Dollár, P., Girshick, R., He, K., Hariharan, B., & Belongie, S. (2017). Feature pyramid networks for object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 2117-2125).

[29] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).

[30] Lin, T. Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., ... & Zitnick, C. L. (2014, September). Microsoft coco: Common objects in context. In European conference on computer vision (pp. 740-755). Springer, Cham.

[31] MMPose Contributors: OpenMMLab Pose Estimation Toolbox and Benchmark, https://github.com/open-mmlab/mmpose (2020)

[32] Sun, K., Xiao, B., Liu, D., & Wang, J. (2019). Deep high-resolution representation learning for human pose estimation. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 5693-5703).

[33] Tan, M., & Le, Q. (2019, May). Efficientnet: Rethinking model scaling for convolutional neural networks. In International conference on machine learning (pp. 6105-6114). PMLR.

SALIENT SIGN DETECTION IN SAFE AUTONOMOUS DRIVING: AI WHICH REASONS OVER FULL VISUAL CONTEXT

Ross Greer
Akshay Gopalkrishnan
Nachiket Deo
Akshay Rangesh
Mohan Trivedi
Laboratory for Intelligent & Safe Automobiles[1]
University of California San Diego
USA

Paper Number 23-0333

ABSTRACT

Detecting road traffic signs and accurately determining how they can affect the driver's future actions is a critical task for safe autonomous driving systems. However, various traffic signs in a driving scene have an unequal impact on the driver's decisions, making detecting the salient traffic signs a more important task. Our research addresses this issue, constructing a traffic sign detection model which emphasizes performance on salient signs, or signs that influence the decisions of a driver. We define a traffic sign salience property and use it to construct the LAVA Salient Signs Dataset, the first traffic sign dataset that includes an annotated salience property. Next, we use a custom salience loss function, Salience-Sensitive Focal Loss, to train a Deformable DETR object detection model in order to emphasize stronger performance on salient signs. Results show that a model trained with Salience-Sensitive Focal Loss outperforms a model trained without, with regards to recall of both salient signs and all signs combined. Further, the performance margin on salient signs compared to all signs is largest for the model trained with Salience-Sensitive Focal Loss.

INTRODUCTION

Detecting and recognizing traffic signs is an important module for an autonomous vehicle to observe and interact with its surroundings in a safe manner. The Safety of the Intended Functionality (SOTIF) process [1] examines highly automated systems for possible hazards and triggering events for unintended behaviors; in this framework, failure to detect a sign crucial to driving performance would be considered a triggering event, independent of the hazardous events, based on system limitations. Accordingly, detection systems are continuously improved to push the safe limits of their operation. Until recently, standard object detectors operated by proposing regions of interest or considering a standard set of anchors or window centers within an image, and classifying the contents of the found region. These approaches are typically limited by the span of the convolutional filters which drive them; these filters operate on local windows, or with a pre-determined span and spacing. While the reach of the convolutional filters can be tuned to spread and cover the entire image, doing so creates massive computational costs or creates gaps in coverage. As a solution, the popular transformer model has been proposed as a means of reasoning over the entire image and bringing forward features relevant to the region of interest. To minimize computational costs, this approach has been further refined to include a stage of learning (via a limited number of deformable attention heads) where an image should be sampled to extract meaningful relational features to the region of interest. This approach is known as the Deformable Detection Transformer, introduced in technical detail in the following section.

[1] cvrr.ucsd.edu

While advances in detection may improve sign recall, we pose one more consideration to be addressed in driving scenes: many signs simultaneously compete for the attention of a human driver or autonomous driving system. While the ideal intelligent vehicle detection module will have perfect precision and recall of all signs in the field of view, environmental noise and underrepresented examples make it possible that detectors continue to make mistakes. However, on the assumption that error is unavoidable, there are some errors preferred over others. For example, it is less critical that a vehicle passing by a freeway exit sees the sign corresponding to the speed limit of an off-freeway side street, or that a vehicle in the right lane preparing to make a right-hand turn sees the lane guidance for the left lane to navigate the intersection. We ascribe this quality of pertinence and attention-worthiness to the word *salience*, as introduced in [2]. We define the term as follows, with clarification on edge cases further described in Methods section:

Salience

A sign is salient if it has the potential to directly influence the next immediate decision to be made by the ego vehicle if no other vehicles were present on the road. Additionally, for signs directing traffic by lane, only signs pertaining to the lane the ego vehicle is in can be classified as salient. In the case of multiple sequential intersections or highway exits visible in the same frame, only signs pertaining to the next immediate intersection or exit could be labeled as salient.

Recent research in sign salience has shown that factors such as sign location, sign appearance, road type, and planned vehicle maneuver can be used to classify signs by salience [2]. Here, we propose a benefit of sign data with salience annotations: salience-aware training methods can be used to improve training of sign detection systems. We make three contributions: (1) creation of the large, salience-annotated LAVA Salient Signs Dataset, (2) definition of Salience-Sensitive Focal Loss, and (3) experimental evaluation of the impact of Salience-Sensitive Focal Loss while training detection transformer models.

RELATED RESEARCH

Traffic Sign Detection and Classification Models

Traffic sign detection has been well addressed by the field such that near perfect sign detection can be achieved on public sign datasets like the German Traffic Sign Detection Benchmark [3] and similar benchmarks. Detecting traffic signs requires cameras monitoring traffic scenes, which allow us to extract frames from videos and build traffic sign annotation datasets. Trivedi et al. [4] proposed that the best way to capture this traffic surveillance is through a multicamera surveillance approach known as distributed interactive video arrays (DIVA). DIVA helps address issues single view cameras have like handling occlusion and having many overlapping views to obtain 3D information. Such a multicamera system can facilitate easier traffic sign detection by addressing the issues mentioned. Some examples of high performance of traffic sign detection on public traffic sign datasets include:

- Using a separate traffic sign detector model and then a sign recognition model [5]. The traffic sign detection model learns the color of the sign and then the shape, and the sign recognition model works best with an ensemble of CNNs.
- A fully convolutional network to guide traffic sign proposals and then a CNN for sign classification [6]. The FCN learns the rough regions of where the traffic signs are present and the CNN identifies the traffic signs and removes false positives with non-max suppression.
- A Pyramid Transformer that uses atrous convolutions and a RCNN as a backbone [7]. This approach improves the network's ability to detect traffic signs of various sizes.
- Using transfer learning with state-of-the-art object detection models on the German Traffic Sign Detection Benchmark dataset [8]. Faster R-CNN Inception Resnet V2 achieves the best mean average precision while R-FCN Resnet 101 has the best tradeoff between accuracy and execution time.

Transformers have begun to outperform other deep learning techniques like CNNs since they can reason over full image context, or learn where to look to extract more features from an image. The detection transformer DETR [9] is a transformer that allows to learn such global image context and achieves state-of-the-art performance on the COCO object detection dataset. DETR is an end-to-end object detection module that treats object detection as a direct set prediction problem and removes the need for any hand-designed components used by other object detection models. A main weakness of DETR is that it has low performance on detecting small objects. Deformable DETR [10] builds on DETR, reducing the computational complexities and also improving performance on detecting small objects. Deformable DETR uses a different attention module that focuses on a subset of sampling points to perform object detection. This method shows theoretical promise in situations where novel, unusual, or newly emergent signs may appear [11], as the signs can be detected not only on the contents of a box which anchors and tries to recognize the sign's face pattern, but also through inferring on learned generic, face-independent contextual features from training. In this work, we apply this state-of-the-art object detection module to the application of traffic sign detection. In addition, we show that we can steer Deformable DETR to improve performance on salient signs via a novel loss function.

Traffic Sign Datasets

There are various traffic sign datasets that allow for researchers to develop traffic sign detection and classifications dataset. A comparison of the size and features of many traffic sign datasets can be seen in [2]. For this paper, we extend the LISA Amazon-MLSL Vehicle Attributes Dataset (LAVA) [12] to create the LAVA Salient Signs (LAVA SS) Dataset, the only dataset which includes the salience property we are interested in utilizing. The datasets and their important properties are listed in the table below:

Table 1

Comparison of Traffic Sign Datasets. The LAVA Salient Signs (LAVA SS) Dataset is used for our research and is the only dataset in this table to include the salience property for traffic signs.

Dataset	Number of Images	Important Features
LISA Traffic Sign Dataset [13]	7,855	occlusion, on-side road
LISA Amazon-MLSL Vehicle Attributes Dataset [12]	14,112	10s video context, occlusion, salience
LAVA Salient Signs Dataset	31,191	10s video context, occlusion, **validated salience**

Traffic Object Salience Research

Learning to focus on salient vehicle objects and construct vehicle visual attention mechanisms has been studied by various researchers, with many using different definitions of what it means to be a "salient" traffic object. We categorize two main types of object saliency from related research: instructive and attentive salience. Attentive salience relates to what objects and directions drivers tend to look at even if these objects may not be what a driver should look at. For this definition of salience, it is often important to monitor the driver's eye gaze to estimate where they are looking at. Taware and Trivedi [14] use driver pose dynamic information to determine the likelihood of a driver gaze zone. This approach tracks facial landmarks like eye corners, nose tip, and nose corners to determine head pose and use the pose to predict the gaze estimation. They found using head pose dynamic features over time increased performance versus using static features like current head pose angles. Robust attentive salience systems must be invariant to different subjects, scales, and perspectives. Vora et al. [15] address this gaze generalization issue using a convolutional neural network to predict driver gaze direction. To improve generalization, they collected a large naturalistic dataset that used ten different subjects and was tested on three unseen subjects. Dua et al. [16] create the first large-scale driver gaze mapping dataset DGAZE, allowing to study attentive salience and where drivers tend to look at for different road and traffic conditions. This dataset contains data from a lab setting of road and driver camera views. Pal et al. [17] learn attentive salience by developing a model named SAGE-Net that uses attention mechanisms to learn how to predict an autonomous vehicle's focus of attention. SAGE-NET uses driver gaze and other important properties like the distance to objects and ego vehicle speed to determine object saliency. Tawari et al. [18] represent gaze behavior for a sequence of image frames by constructing a saliency map using a fully convolutional RNN. The saliency map uses three kinds of pixels: salient (positive) pixels, non-salient (negative) pixels, and neutral pixels. Other than gaze estimation, other important factors like predicting driver maneuvers and braking intent are important to understand attentive salience. Ohn-Bar et al. [19] use a multi-camera head pose estimation model to predict overtaking and braking intent and maneuvers. This system emphasizes real-time performance, which is critical for any attentive salience model in order to timely observe the driver state and react if they are distracted.

In contrast to attentive salience, instructive salience aims to emphasize important objects which an ego vehicle should observe and respond to; these objects should influence the car's future decisions. Our work focuses on instructive saliency and highlights what traffic signs the car needs to be aware of to safely operate. Instructive salience models are often more costly since labeling important objects requires understanding how various road objects and signs influence a driver's decisions and vehicle navigation, so a cognitively-demanding and maneuver-aware manual process is required to annotate such data. To overcome these challenges, Bertasius et al. [20] use an unsupervised learning approach to learn how to detect important objects in first-person images without any instructive salience labels, skipping the costly manual annotation process. The unsupervised network uses a segmentation network to propose possible important objects, and this output is fed into a recognition agent which uses these proposals and other spatial features to predict the important objects. Greer et al. [2] utilize a supervised learning process to classify salient road signs, which can be applied for efficient dataset annotation in future road sign data collection after initial training. Lateef et al. [21] use a conditional GAN to predict what a driver should be looking at in a traffic scene, which parallels this instructive salience definition. For constructing ground truths, they use semantic labels (annotations of traffic objects in images) from various autonomous driving datasets and use various saliency detection algorithms that select which object annotations are the most important. Zhang et al. [22] use interaction graphs to perform object importance estimation in driver scenes. The interaction graph updates features of each object node through interactions with graph convolutions. This task learns to model instructive saliency, as Zhang et al. note that their object importance definition relates to how objects can help with the driver's real-time decision making and improve safe autonomous driving systems.

METHODS

Data Collection

The LISA Amazon-MLSL Vehicle Attributes (LAVA) dataset contains labeled bounding boxes of traffic signs taken from a front-facing camera of a vehicle. This dataset was collected from the greater San Diego Area and contains a variety of road types, lighting, road types, and traffic conditions. The traffic signs are categorized as stop, yield, do not enter, wrong way, school zone, railroad, red and white regulatory, white regulatory, construction and maintenance, warning, no turn, one way, no turn on red, do not pass, speed limit, guide, service and recreation, and undefined. Along with the traffic sign categorization, we carefully labeled the sign salience property for all the sign annotations. A sign salience validation process was also performed in which the salience property for a sign annotation was checked again for consistency with the above definition of salience; the resulting dataset is referred to as the LAVA Salient Signs (LAVA SS) Dataset. This data collection process ensured that the salience property was properly labeled and the curated dataset had accurate ground truths. In the process of annotation, the provided definition of salience was used as a standard for annotation, with select frequent ambiguous cases handled according to the additional criteria below:

- Guides (signs which indicate the street name, often green) at intersections and freeways were labeled as salient, as long as such signs were the closest such guide in the scene. That is, in the case of multiple sequential intersections, only the guides of the nearest intersection to the car would be labeled as salient. Salient guides should be visible to the vehicle and indicate a possible street the car could turn onto or an exit the car could take. We note that guides tend to have the highest annotator ambiguity, as the class "guide" contains instances of street-level guides as well as freeway-level guides, which may be interpreted differently by different annotators. Likewise, parking guides are a highly missed ground-truth annotation. For this reason, certain applications may benefit from computing precision disregarding guides and parking signs, especially since such signs are less safety critical.

- Instructions pertaining to HOV or Carpool Lanes are marked as salient when the vehicle is moving in the direction of traffic, regardless of lane. Such a sign may indicate a lane available to the intelligent vehicle for optimized traffic flow, or a lane which the vehicle is required to leave if requirements are not met.
- A "No Parking" sign is salient only if the ego vehicle is in a lane which has immediate access to the restricted parking location (e.g. far right lane). An example of this rule is shown in Figure 1.
- While most signs which are facing backwards are marked as non-salient (since they provide instruction to an oncoming lane), in some cases, a yellow reflective warning sign is placed on the back of the sign. In these cases, we mark this as a salient warning sign if the sign is adjacent to the ego vehicle's lane. The vehicle should be aware of such signs to avoid collision with the sign or median. This rule is exemplified in Figure 2.
- Signs which indicate a fine for littering or carpool violations are regarded as non-salient, since an intelligent vehicle should not be littering under any circumstance, nor motivated by the cost of breaking a traffic ordinance.

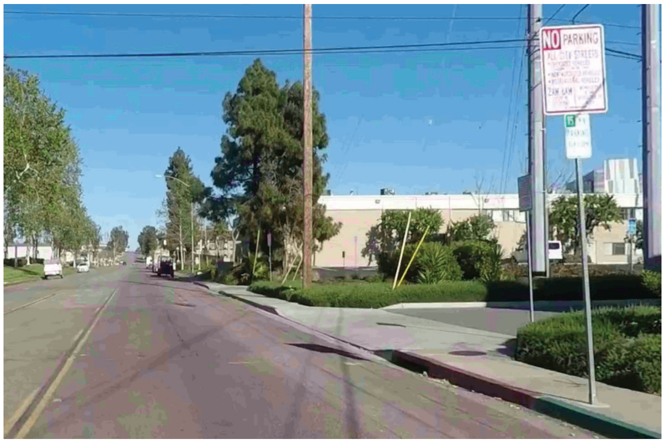


Figure 1. Because this No Parking sign is located in the lane closest to the ego vehicle, it is considered salient. Were the ego vehicle in the left lane of a two lane road, this would be annotated as non-salient.

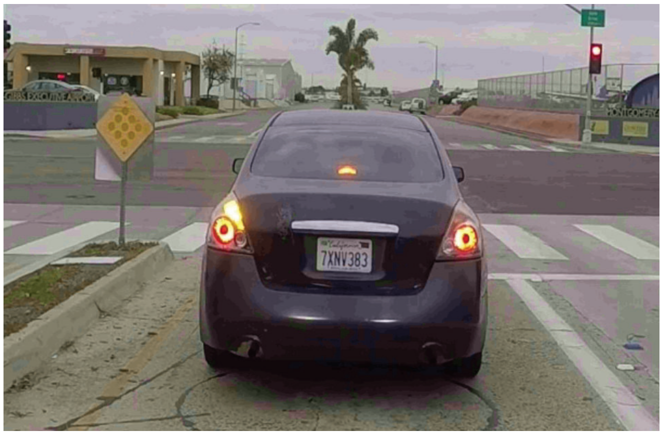


Figure 2. This sign is facing away from the ego vehicle, but because the reflector is placed to warn the vehicle of its presence, it is annotated as salient.

Example sign annotations from the LAVA Salient Signs dataset are shown in Figure 3. The LAVA Salient Signs dataset contains 31,992 sign annotations with 20,377 annotations being salient and 11,615 annotations being non-salient. The sign type frequencies for the LAVA Salient Signs Dataset are defined in Figure 4. Because the data was collected and annotated using a selection method which promotes maximal coverage of driving area (including diversity of driving environment, conditions, and road types), the non-uniform distribution of signs may reflects the real-world distribution of salient and non-salient signs as well as the real-world distribution of sign categories during naturalistic driving.

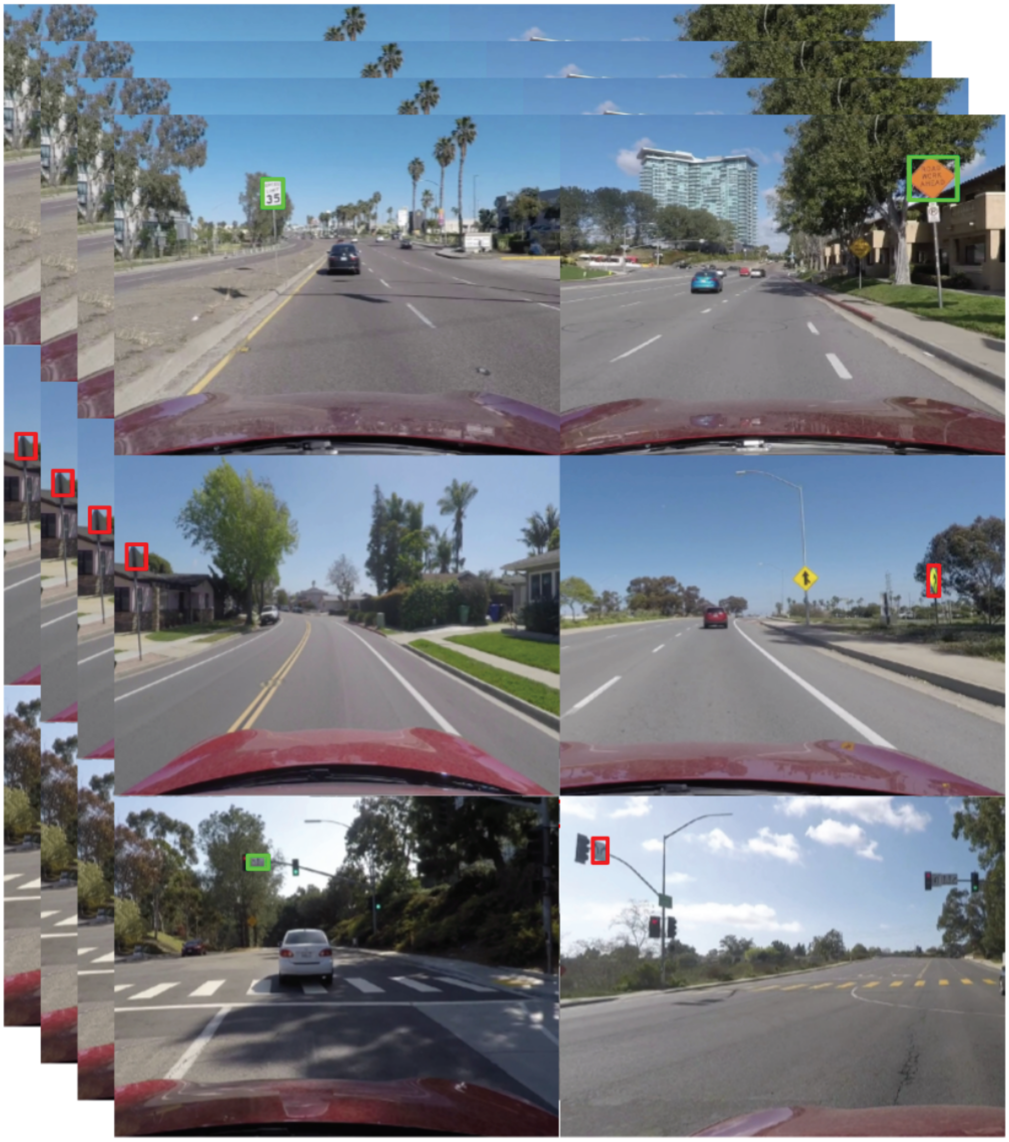


Figure 3. Example Sign Annotations from the LAVA Salient Signs Dataset. A green bounding box indicates a salient sign and a red bounding box indicates a non-salient sign. As shown in the figure, the salient annotations often mean that the sign relates to the current lane or intersection the driver is in and provides meaningful information that affects the driver's future actions. On the other hand, the non-salient signs are often in a different lane, intersection, or face the wrong way, so these signs don't offer any important information. A vehicle's intended maneuver is important in classifying sign salience, so temporal dynamics should be considered when annotating and utilizing salience data, as explained in [2].

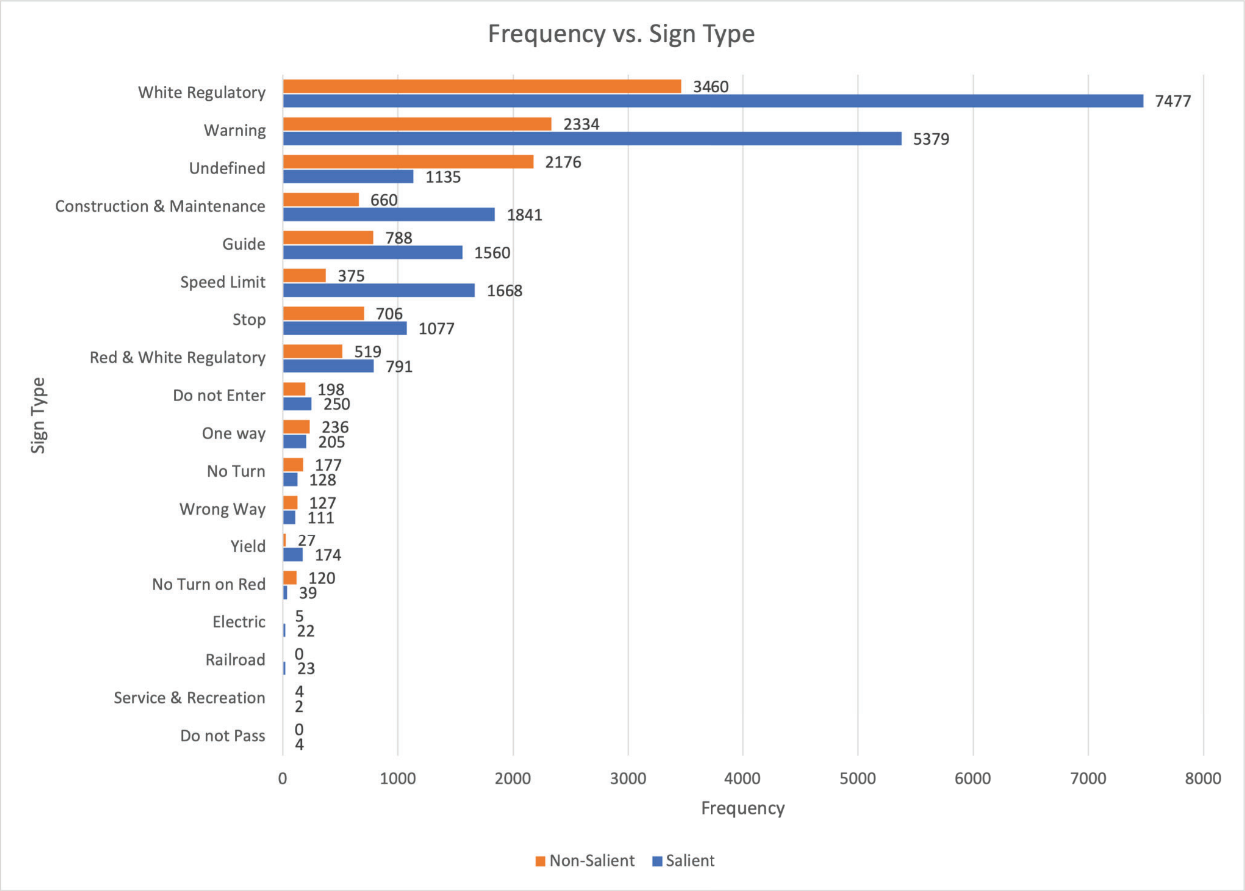


Figure 4: Sign Type Frequencies in the LAVA Salient Signs Dataset. The blue columns are for salient signs and the orange for non-salient signs. The White Regulatory, Construction & Maintenance, and Warning Signs were the most common sign types. This distribution of signs may be dependent on location, as all of our data was collected in the greater San Diego area.

Sign Detection with Deformable DETR

We use Deformable DETR, introduced in the Related Works section, to detect signs in the images. This detection method forms our performance baseline, described by Figures 2 and 3. We split the LAVA Salient Signs Dataset into 25,591 training instances, 3,200 validation instances, and 3,201 test instances. The model is trained for 15 epochs, retaining the model which reports the strongest precision (with a "hit" at 0.5 intersection-over-union, and 100 maximum detections per image). We use a ResNet50 backbone [23], 300 attention heads, a learning rate of 0.0002, a batch size of 2, and employ gradient clipping and learning rate decay.

Prioritizing Salient Signs with Salient-Sensitive Loss

The bounding box regression module of each Deformable DETR detection head is a 3-layer feed-forward neural network. Each detection head also has one linear projection for classification of the estimated bounding box into categories of foreground (object) or background (no object). This classification is trained using a sigmoid focal loss [24], an extension of standard categorical cross-entropy which down-weights easy examples to focus training on hard negatives. The equation for focal loss is

$$FL(p_t) \; = \; -\alpha_{FL}(1-p_t)^{\gamma} log(p_t), \qquad \text{(Equation 1)}$$

where α_{FL} is a hyperparameter to balance the focal loss among other loss functions, γ is a focusing parameter to control the influence of hard negatives, and p_t is the predicted probability associated with the ground truth class.

As explained in the introduction, the goal of our detection model is to prioritize successful detection on signs which are salient, ideally placing any model error on non-salient signs. To achieve this, we weigh the focal loss heavily for salient signs according to the function

$$FL(d, p_t) \;=\; -\alpha_{FL} w_{SS}(d)(1 - p_t)^{\gamma} log(p_t) \qquad \text{(Equation 2)}$$

where $w_{ss}(d) = \alpha_{ss}$ if the ground truth sign nearest detection d is salient, and $w_{ss}(d) = 1$ otherwise. In our case, we use a hyperparameter $\alpha_{ss} = 4$. We name the function $FL(d, p_t)$ ***salience-sensitive focal loss***.

RESULTS

The performance of the Deformable DETR model on the LAVA Salient Signs Dataset with and without salience-sensitive focal loss is provided in Figures 5-7. These figures display interpolations between precision-recall pairs generated with a detection thresholding of 0 to 1 in increments of 0.1 (with an early stop at thresholds where no positive detections are made). We note that thresholds should be tuned for precision and recall according to intended application; a representative descriptor of performance is given by the precision-recall curves. Results suggest that not only does training with salience-sensitive focal loss distribute error to non-salient signs instead of salient signs, but that the method actually improves overall performance of the model under otherwise equal training.

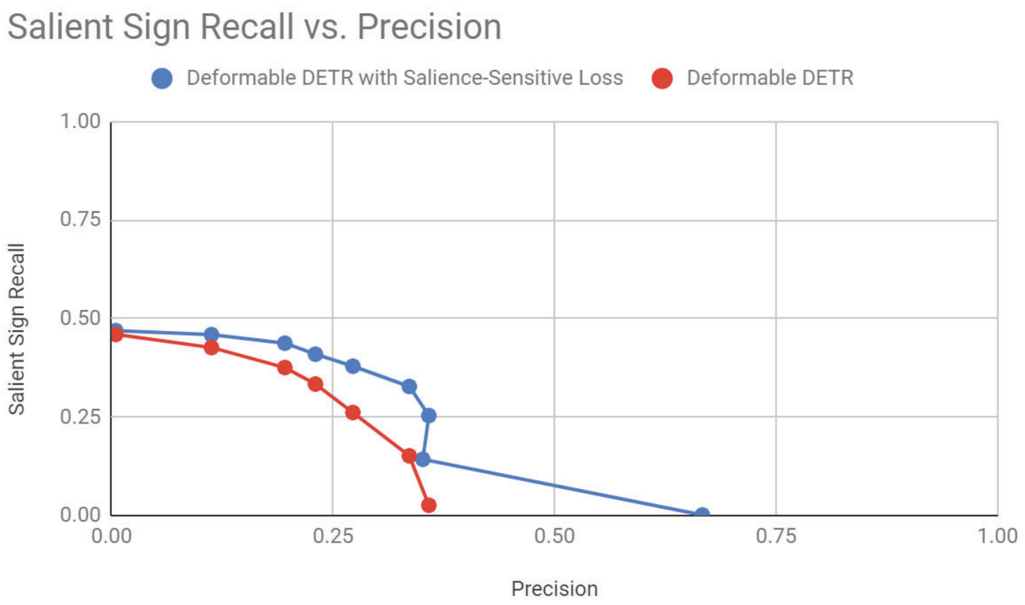


Figure 5. Deformable DETR shows uniformly better performance in recalling salient signs when using Salience-Sensitive Focal Loss. Additionally, as a general measure of performance, the area under the precision-recall curve is greater when using Salience-Sensitive Loss.

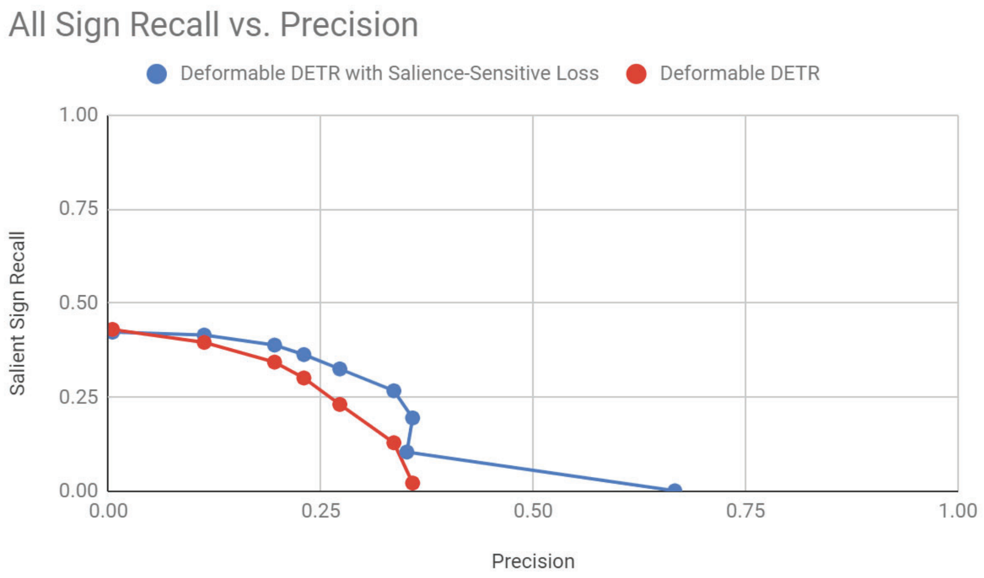


Figure 6. Deformable DETR additionally shows better performance in recalling all signs (both salient and non-salient) when using Salience-Sensitive Focal Loss. A possible reason for this improvement is that signs which are salient tend to be localized to particular image regions which amass both sign types, whereas some locations of non-salient signs would very rarely have a salient sign appear. This may help guide the transformer as it learns which regions of the image to attend.

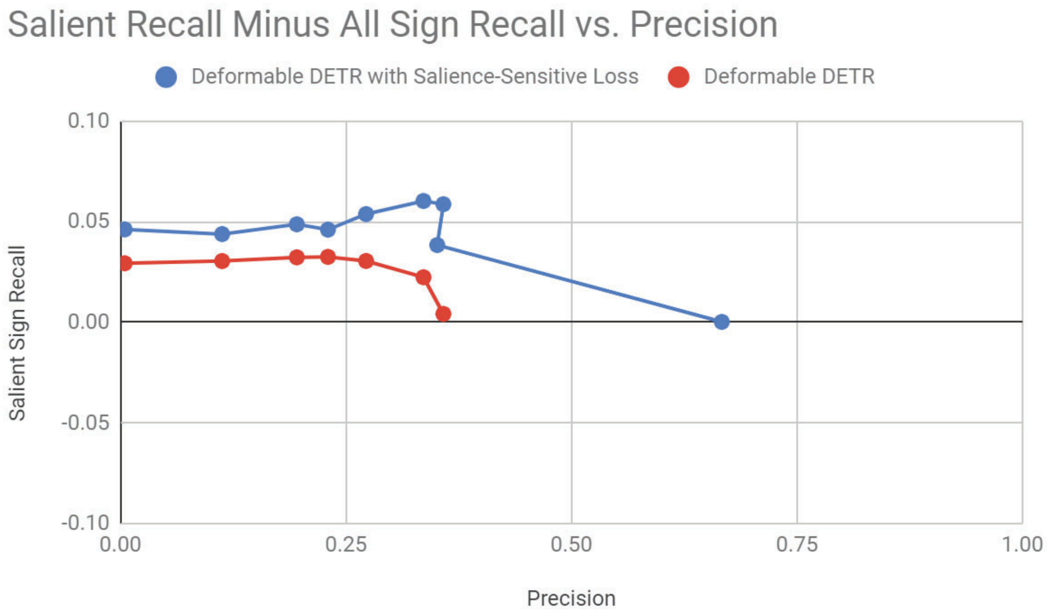


Figure 7. How well does the Salience-Sensitive Loss bring out performance on salient signs? In this graph, we show the difference in performance between salient sign recall and all sign recall (in other words, how much better is the model at recalling salient signs than the aggregate collection of signs). Deformable DETR does generally perform better on salient signs than all signs together, but with exception as precision increases (in fact, negative at its greatest precision). On the other hand, Deformable DETR with salience-sensitive focal loss maintains improved performance on salient signs, and at greater margin than the baseline model.

CONCLUDING REMARKS

Detection transformers make use of full-image context in a selective manner, and this property makes them an excellent candidate for tasks which often require human drivers to make evaluations over which portion of their visual field to attend to. We illustrated the performance of the recent (and computationally tractable) Deformable DETR model on sign detection for a large dataset even under limited computational budget, providing a baseline for model performance on the dataset. Preliminary results are provided under reduced training time to illustrate the potential of detection-transformer-based methods and to provide a clear demonstration of the impact of modified loss functions on model performance compared to a baseline. Under elongated training regimens and increased dataset sizes, sign detection modules would reasonably be expected to perform to the standards of comparable benchmark models and datasets as described in related research.

We expand this analysis, noting that road objects carry an implicit importance and relevance to the ego vehicle. By including this property, salience, in the training regimen, we show that the sign detector can be further improved, both in general performance and especially in recall of signs which are most important to the safe operation of the autonomous vehicle. Gains in sign detection performance afforded via modification of the training loss function, especially in recalling salient signs, are directly related to the safety of the vehicle in navigating a scene and responding appropriately and safely to surrounding agents.

ACKNOWLEDGEMENTS

We are grateful to LISA research sponsors, especially to Dr. Suchitra Sathyanarayana, Ninad Kulkarni, and Jeremy Feltracco of AWS Machine Learning Solutions Laboratory for sharing AWS Sagemaker GroundTruth[2] and LISA colleagues for their assistance in creating the novel LAVA dataset and its derivatives.

REFERENCES

[1] Rau, P., Becker, C., & Brewer, J. (2019). Approach for deriving scenarios for safety of the intended functionality. In Proc. ESV (pp. 1-15).

[2] Greer, R., Isa, J., Deo, N., Rangesh, A., & Trivedi, M. M. (2022). On Salience-Sensitive Sign Classification in Autonomous Vehicle Path Planning: Experimental Explorations with a Novel Dataset. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (pp. 636-644).

[3] Stallkamp, J., Schlipsing, M., Salmen, J., & Igel, C. (2012). Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition. *Neural networks*, *32*, 323-332.

[4] Trivedi, M. M., Gandhi, T. L., & Huang, K. S. (2005). Distributed interactive video arrays for event capture and enhanced situational awareness. *IEEE Intelligent Systems*, (pp 58-66).

[5] A. Vennelakanti, S. Shreya, R. Rajendran, D. Sarkar, D. Muddegowda & P. Hanagal. Traffic Sign Detection and Recognition using a CNN Ensemble. 2019 IEEE International Conference on Consumer Electronics (ICCE) (pp. 1-4).

[6] Zhu, Y., Zhang, C., Zhou, D., Wang, X., Bai, X., & Liu, W. (2016). Traffic sign detection and recognition using fully convolutional network guided proposals. Neurocomputing, 214, 758-766.

[7] O. N. Manzari, A. Boudesh & S. B. Shokouhi. (2022). Pyramid Transformer for Traffic Sign Detection. 2022 12th International Conference on Computer and Knowledge Engineering (ICCKE) (pp. 112-116).

[8] Arcos-Garcia, A., Alvarez-Garcia, J. A., & Soria-Morillo, L. M. (2018). Evaluation of deep neural networks for traffic sign detection systems. *Neurocomputing*, *316*, 332-344.

[9] Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., & Zagoruyko, S. (2020, August). End-to-end object detection with transformers. European Conference on Computer Vision (pp. 213-229).

[2] https://aws.amazon.com/blogs/machine-learning/creating-a-large-scale-video-driving-dataset-with-detailed-attributes-using-amazon-sagemaker-ground-truth/

[10] Zhu, X., Su, W., Lu, L., Li, B., Wang, X., & Dai, J. (2021). Deformable DETR: Deformable transformers for end-to-end object detection.

[11] Koopman, P., Hierons, R., Khastgir, S., Clark, J., Fisher, M., Alexander, R., ... & McDermid, J. A. (2019). Certification of highly automated vehicles for use on uk roads: Creating an industry-wide framework for safety.

[12] Kulkarni, N., Rangesh, A., Buck, J., Feltracco, J., Trivedi, M., Deo, N., Greer, R., Sarraf, S., & Sathyanarayana, S. (2021). Create a large-scale video driving dataset with detailed attributes using Amazon SageMaker Ground Truth.

[13] Møgelmose, A., Liu, D., & Trivedi, M. M. (2014, October). Traffic sign detection for us roads: Remaining challenges and a case for tracking. *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)* (pp. 1394-1399).

[14] Tawari, A., & Trivedi, M. M. (2014, June). Robust and continuous estimation of driver gaze zone by dynamic analysis of multiple face videos. *2014 IEEE Intelligent Vehicles Symposium Proceedings* (pp. 344-349).

[15] Vora, S., Rangesh, A., & Trivedi, M. M. (2017, June). On generalizing driver gaze zone estimation using convolutional neural networks. *2017 IEEE Intelligent Vehicles Symposium (IV)* (pp. 849-854).

[16] Dua, I., John, T. A., Gupta, R., & Jawahar, C. V. (2020, October). Dgaze: Driver gaze mapping on road. *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 5946-5953).

[17] Pal, A., Mondal, S., & Christensen, H. I. (2020). " Looking at the right stuff"-Guided semantic-gaze for autonomous driving. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 11883-11892).

[18] Tawari, A., Mallela, P., & Martin, S. (2018, November). Learning to attend to salient targets in driving videos using fully convolutional rnn. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)* (pp. 3225-3232).

[19] Ohn-Bar, E., Tawari, A., Martin, S., & Trivedi, M. M. (2015). On surveillance for safety critical events: In-vehicle video networks for predictive driver assistance systems. *Computer Vision and Image Understanding*, *134* (pp. 130-140).

[20] Bertasius, G., Soo Park, H., Yu, S. X., & Shi, J. (2017). Unsupervised learning of important objects from first-person videos. *Proceedings of the IEEE International Conference on Computer Vision* (pp. 1956-1964).

[21] Lateef, F., Kas, M., & Ruichek, Y. (2021). Saliency heat-map as visual attention for autonomous driving using generative adversarial network (GAN). *IEEE Transactions on Intelligent Transportation Systems*.

[22] Zhang, Z., Tawari, A., Martin, S., & Crandall, D. (2020, May). Interaction graphs for object importance estimation in on-road driving videos. *2020 IEEE International Conference on Robotics and Automation (ICRA)* (pp. 8920-8927).

[23] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).

[24] Lin, T. Y., Goyal, P., Girshick, R., He, K., & Dollár, P. (2017). Focal loss for dense object detection. In Proceedings of the IEEE international conference on computer vision (pp. 2980-2988).