It’s all about you: Personalized in-Vehicle Gesture Recognition with a Time-of-Flight Camera

Amr Gomaa*
German Research Center for Artificial Intelligence (DFKI)
Saarbrücken, Germany
Saarland Informatics Campus
Saarbrücken, Germany
amr.gomaa@dfki.de

Guillermo Reyes*
German Research Center for Artificial Intelligence (DFKI)
Saarbrücken, Germany
guillermo.reyes@dfki.de

Michael Feld
German Research Center for Artificial Intelligence (DFKI)
Saarbrücken, Germany
michael.feld@dfki.de

ABSTRACT

Despite significant advances in gesture recognition technology, recognizing gestures in a driving environment remains challenging due to limited and costly data and its dynamic, ever-changing nature. In this work, we propose a model-adaptation approach to personalize the training of a CNNLSTM model and improve recognition accuracy while reducing data requirements. Our approach contributes to the field of dynamic hand gesture recognition while driving by providing a more efficient and accurate method that can be customized for individual users, ultimately enhancing the safety and convenience of in-vehicle interactions, as well as driver’s experience and system trust. We incorporate hardware enhancement using a time-of-flight camera and algorithmic enhancement through data augmentation, personalized adaptation, and incremental learning techniques. We evaluate the performance of our approach in terms of recognition accuracy, achieving up to 90%, and show the effectiveness of personalized adaptation and incremental learning for a user-centered design.

CCS CONCEPTS

• **Human-centered computing** → **User models**; **Gestural input**; • **Computing methodologies** → **Object recognition**; **Neural networks**.

KEYWORDS

Incremental Learning; Gesture Recognition; User-specific Adaptation; Personalized Models; Deep Learning

1 INTRODUCTION

The last decade has brought significant breakthroughs in speech recognition, image recognition, semantic segmentation, and many other domains. These breakthroughs have been significantly due to the advancement in deep learning (DL) techniques. In particular, this is due to more powerful computing hardware and larger datasets [24], but also new ideas and architectures [54]. Gesture recognition is one of these domains. Gestures are a natural form of human interaction, but teaching machines to recognize gestures (particularly dynamic ones) can be challenging despite these advances. Some of these challenges relate to the technology used to record the data. Although RGB cameras are commonly used in gesture recognition, they are not optimal for the dynamic situation of in-vehicle interaction. That is because they cannot handle poor lighting conditions (e.g., at night) and hands’ high-speed motion, which require capturing devices with high frame rates.

Alternatively, time-of-flight (ToF) depth cameras do not suffer from these problems and can be used with a high frame rate in day and night conditions. On the other hand, ToF cameras can produce noisy results, where the depth of individual pixels changes slightly across frames, or some information is lost. Another problem is that DL architectures are known to require large amounts of data. This is primarily due to the high number of parameters required to train an accurate model. The data collection process and resources needed to train a DL model can be extremely costly. Some techniques, like transfer learning, can help reduce the data needed to train a DL model by adapting a pre-trained model from a similar task to a new one. However, its effectiveness depends on how similar the data from both tasks are. In this paper, we are considering the specific use cases of in-vehicle Human Machine Interaction(HMI) gesture interaction using a ToF camera. The first challenge is the lack of datasets for the specific camera view in an in-vehicle environment for hand gesture recognition. Another challenge is that, as far as we know, in-vehicle hand gesture recognition datasets utilize an RGB camera instead of a ToF one. Since there are essential differences between RGB and depth data, traditional transfer learning is not directly feasible.

Furthermore, even in the best-case scenario in which it is possible to train a DL dynamic gesture recognition model using ToF cameras with little data, there will always be individual differences in how users perform the gestures. This, in turn, causes the gestures of some individuals not to be as well recognized as those of others. For this reason, gathering data pertaining to each user’s individual differences to train personalized models is essential. We gather inspiration for personalization from recent studies on the topic of human-centered artificial intelligence (HCAI), which is gaining rapid and significant interest among researchers in both artificial intelligence (AI) and human-computer interaction (HCI) [6, 38, 50, 58]. Finally, this paper tackles the previously mentioned challenges using a Convolutional Long-Short Term Memory Neural Network (CNNLSTM) [57] and several user-specific adaptations and incremental learning techniques. In particular, we first collect a new dynamic hand gesture data set inside a car using a ToF camera. Then, we highlight the effect of data augmentation techniques on enhancing the model’s accuracy and present different incremental learning adaptation strategies for user-centered gesture recognition. Thus, **this paper’s contribution has several folds, as follows**. 1) We study the feasibility of hand gesture recognition for in-vehicle

[*] Both authors contributed equally to this research.

interaction as conceptualized by modern car manufacturers using a ToF camera instead of RGB; 2) we propose several essential reproducible preprocessing techniques for ToF cameras as a guideline for future use that is applicable for the automotive domain specifically and other domains generally; 3) we exploit individual differences in gesture performance by utilizing several personalization and model adaptation techniques such as transfer learning, data augmentation, and incremental learning to enhance the recognition accuracy.

2 RELATED WORK

The first form of communication we learn as infants is hand gestures, even before learning to speak, which is why gestures are the most natural form of communication used by humans [56]. Furthermore, there are around seven thousand languages spoken in the world [5]. However, simple hand gestures (e.g., mid-air gestures, pointing, waving, etc.) are a common form of communication among people, making it more understandable for machines. Therefore, researchers have tried to incorporate hand gestures into various domains [9, 18, 20, 21, 23, 27, 37, 40, 43, 51, 59, 60]. Specifically for the automotive domain, researchers attempted to control the infotainment and various parts inside the vehicle [4, 7, 12, 31, 32, 35, 39, 41, 46, 47, 53, 62] as well as interact with objects outside the vehicle [3, 13, 15–17, 33, 48]. They chose hand gestures for several reasons such as: the simplicity and naturalness of hand gestures when interacting with a somewhat complicated machine like a modern car; and the reduced cognitive load on the user when using hand gestures while driving a vehicle (which should be the main focus of the driver).

Several researchers have investigated hand tracking and gesture recognition for more than 30 years. Zimmerman et al. [61] created a hand glove augmented with analog flux sensors to measure finger bending and track simple gesture interactions. Takahashi and Kishino [55] conducted several studies using this glove-based device to determine the most commonly used gestures by users. Subsequently, several researchers conducted similar studies for glove-based gesture recognition, as highlighted by Dipietro et al. [10] in their survey. However, these approaches suffer from several problems, such as the need to wear intrusive gadgets, sensor failures, and inaccuracies. Alternatively, researchers attempted vision-based approaches for hand gesture recognition. Min et al. [30], Rigoll et al. [45], Eickeler et al. [11], and Chen et al. [8] utilized Hidden Markov Models (HMMs) for training and accurate classification of the desired hand symbol (i.e., gesture) under stationary background and fixed light conditions. Similarly, Ren et al. [44], and Huang et al. [19] attempted the same task by utilizing image processing techniques (e.g., Gabor filter) and machine learning approaches (e.g., support vector machine (SVM)) to segregate and classify different hand gestures. However, while these approaches supported real-time recognition, they mainly focused on simple static hand gestures with a clear differentiation among them.

More recently, several vision-based approaches for hand gesture recognition have emerged due to the massive breakthrough in image processing, computer vision, and object recognition using deep neural networks. Stergiopoulou et al. [52] utilized a Feed Forward Neural Network (FFNN) for right-hand gesture recognition using a shape-fitting technique. Similarly, Mang [29] utilized an FFNN to classify hand gestures after applying a preprocessing feature extraction method (e.g., Oriented Histograms) on hand images with no background information (i.e., cutout hand images with black background). Alternatively, Nagi et al. [34], Lin et al. [26], Li et al. [25], and Pinto et al. [42] employed a Convolution Neural Network (CNN) learning approach for the recognition due to its high performance on images for several computer vision applications. However, these approaches focus on static gesture recognition scenarios and suffer from limited background information and heavy irreproducible pre-processing techniques. Thus, Maraqa and Abu-Zaiter [28], Neverova et al. [36], Köpüklü et al. [22], and Molchanov et al. [31] investigated Recurrent Neural Network (RNN) and 3D CNN (instead of the traditional 2D CNN approach) to include time series analysis and dynamic gesture recognition, while Tsironi et al. [57] explored the combination of a CNN and RNN into a new architecture named CNNLSTM (an approach first used by Sainath et al. [49] for voice recognition). In this work, we employ a similar CNNLSTM architecture as the state-of-the-art approach for dynamic gesture recognition and introduce easily reproducible preprocessing techniques for ToF cameras.

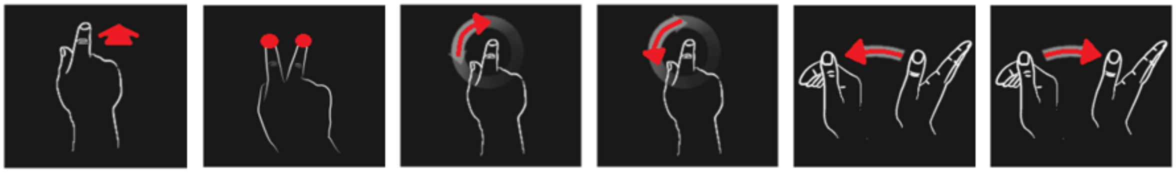
Finally, while previous approaches have addressed many dynamic gesture recognition challenges, they were still lacking in terms of recognition accuracy. We attribute this to two main problems: *Hardware* and *User-specific Variations*. For the hardware part, most of the previous work either used an RGB camera or a low-performance depth camera (i.e., Kinect). In our approach, we utilized a prototype high-resolution 3D time-of-flight camera (i.e., a high-fidelity depth camera) that is already utilized in high-end modern vehicles [1–3]. As for the user-specific variations, most of the current work relies on a one-model-fits-all approach where no adaptation or user-specific personalization are made to the learning model. In contrast, we take inspiration from the Human-centered artificial intelligence (HCAI) domain [6, 38, 50, 58], the incremental learning domain [14], and the transfer learning techniques to employ a user-specific personalized approach that is adaptable for drivers' individual behavior when performing dynamic hand gestures.

3 DESIGN AND PROCEDURE

In this section, the data acquisition process is highlighted, starting with the environment setup, participants' demographics, and data collection procedure and ending with the detailed description of the final collected dataset, including the split for both the general and personalized models.

3.1 Apparatus

The first challenge of in-vehicle hand gesture recognition data acquisition is the necessary instrumentation and technology. As previously mentioned, existing wearable hand-tracking devices are not practical in highly dynamic driving situations, as well as RGB cameras. On the other hand, ToF-based in-vehicle datasets are not available. Thus, we collect our own data set for this purpose. We utilize a state-of-the-art non-commercial hand-tracking camera prototype specially designed for in-vehicle control. This ToF camera was attached to the ceiling of a real car from the inside. It was



Pointing (PT)

Two Finger (TF)

Rotate Clockwise (RCW)

Rotate Counterclockwise (RCCW)

Swipe Left (SL)

Swipe Right (SR)

Figure 1: The six different gestures performed in our dataset and utilized in the the adapted models and incremental learning.

positioned near the front center, above the gear shift, with a top-down perspective without blocking the view of the windshield. This position corresponds to the position of the gesture camera of recent modern cars such as the *BMW Natural User Interface* [1] and *Mercedes-Benz User Experience (MBUX) Travel Knowledge Feature* [2], which provides a realistic and practical setting applicable to existing in-vehicle environments.

3.2 Participants

We recruited 83 participants (41% female) with a mean age of 25.97 years (SD = 5.97) for the data collection phase. Regarding handedness, 90.4% of participants were right-handed. Regarding the driving experience of the participants, the participants had their driver's license on average for 6.54 years (SD = 5.76). Additionally, we collected participants' height and hand length for their possible effect on hand gesture recognition. The participants had an average height of 176.84 centimeter (SD = 10.79) and an average of 66 (SD = 8.6), 95.26 (SD = 9.62), 105.02 (SD = 10.08), 97.72 (SD = 10.39), and 78.29 (SD = 8.43) millimeters for the thumb, forefinger, middle finger, ring finger and pinkie, respectively.

3.3 Procedure

Participants were asked to perform a series of gestures repeatedly. Each gesture was performed ten times per driver. Gestures were performed randomly to avoid any confounding factors or learning effects. Participants were instructed to keep their hands on the wheel as if they were driving to emulate real driving scenarios. The participants would then move their hands from the steering wheel to the area directly below the camera, perform the gesture, and move the hand back to the steering wheel. An experimenter sat in the back of the vehicle throughout the experiment to start and stop the camera without interruption, influence, or comments on the performance of participants' gestures. Although participants were shown pictures representing the gestures at the beginning, they were not instructed exactly how to perform the gesture to avoid influencing them. This made the recorded data set richer in individual variability and more personalized.

Table 1: Participant split according to two disjunct groups. The data of participants in the UBM group are used to train the unviresal background model, subsequently adapted to the participants in the adaptation (AS) group.

Description	UBM	AS	Total
Baseline	51	21	72
Extended	72	11	83

3.4 Dataset

The final data set consists of six different simple gesture classes. They are *pointing*, *two fingers*, *rotate clockwise*, *rotate counterclockwise*, *swipe left*, and *swipe right* gestures as seen in Figure 1. Additionally, variations to these simple gestures are introduced, such as multiple pointing, swipe, and two-finger gestures after each other and multiple rotations or fractions of rotations in the same direction for the rotate gestures. For all experiments in this work, only simple gestures were used. These simple gestures could be enough to be used for basic infotainment interaction such as raising the volume, skipping songs, etc. However, new gestures could be added using the same methods discussed here for expanded interaction possibilities. This data set was split into baseline and extended data sets. The extended dataset included additional gestures per participant (approximately 50 per gesture class) for eleven participants that could be contacted to test the model's incremental training and adaptation on these participants. Participants were split into two disjunct groups: a Universal Background Model (**UBM**) group and an Adaptation Set (**AS**) group as in Table 1.

Depending on the experiment, participants data were further split within each of these groups into training, validation, and test sets. For a person in the UBM group, the data was split into 80% training and 20% validation. On the contrary, for a person in the AS group, the data were split into 50% further training for adaptation, 20% validation and 30% evaluation. In more detail, Table 2 highlights the number of samples in each split of the data set.

4 METHODOLOGY AND RESULTS

In this section, we describe the experiments performed that improve the recognition rate of the models from random chance to over 90% accuracy. However, first, we describe the pre-processing steps that apply to all experiments. Then, we describe the CNNLSTM architecture used in these experiments. We then describe the training of a baseline model and show how this model can be adapted to specific users. We then dive into additional techniques that further enhance the model's performance before experimenting with the amount of data needed to effectively personalize the trained models and how to do this incrementally.

Table 2: Data split in training validation and evaluation sets in the baseline and extended datasets.

Dataset	Training	Validation	Σ UBM	Adaptation	Adapt Val	Evaluation	Σ AS	Σ
Baseline	2455	610	3065	630	252	378	1260	4325
Extended	3710	926	4636	1326	534	792	2652	7288

4.1 Preprocessing

Deep learning architectures are famous for being end-to-end and do not require hand-crafted features. However, typical deep-learning algorithms also require large amounts of data, from thousands to millions of samples. In the lack of data, it is necessary to simplify the problem of deep learning architecture by performing some simple preprocessing steps. The following describes the preprocessing steps we performed.

4.1.1 Number of Frames. Since the data collected contained variations in the sequence length and sampling rate due to inconsistencies in the network at the recording time, the first step was to standardize the number of frames in each training sequence and the frame rate used to train the model. A frame rate of 12 fps and a maximum sequence length of 70 frames were selected because (a) this reduces the number of parameters that need to be trained, decreasing the amount of data needed to train such parameters, and (b) most sequences had a frame rate above 12 fps and a sequence length below 70 frames. Since neural networks expect a fixed size input, gesture sequences that exceeded the maximum sequence length were truncated, and those that fell below the maximum sequence length were zero-padded. Similarly, gesture sequences that exceeded 12 fps were subsampled to this frame rate. Although it may seem that by altering the dataset in this way, some important information might be lost, making the training of the model more difficult, in reality, most of the dropped information does not contribute much to the classification of the gesture, as contiguous frames are highly correlated. Furthermore, most gestures only take one to two seconds to perform. Removing frames that exceed the max sequence length could remove some noise from the dataset, making the resulting data easier to learn.

4.1.2 Image Processing. Aside from fixing the sequence length and subsampling the gesture sequences to 12 fps, a 3D region of interest was defined between 12 and 500 centimeters in depth. Values outside this range were considered to be errors or irrelevant. The images were then standardized, and a Gaussian blur was applied. Finally, the images were scaled down from 320x240 to 80x60 pixels, reducing the number of features needed by the model while maintaining the distinguishing visual features of the hand and fingers.

4.1.3 Hand Segmentation. An essential final pre-processing step was that of hand segmentation. This allowed the model to finally start learning the hand gestures, as it removed a significant part of the remaining noise in the image. This was done using a simple technique called background subtraction. This consisted of two steps. In the first step, a background image was calculated for each gesture using the first six frames of the frame sequence. The average background image was subtracted from all the following frames. By doing this, pixels where there is no movement are reduced to near-zero values, while those that contain movement (i.e., where the hand gesture is performed) keep relatively large values. After a qualitative visual comparison, this approach showed a more accurate segmented hand compared to existing RGB-based hand segmentation approaches such as the two-frame subtraction technique from Rigol et al. [45] and the three-frame differencing technique from Tsironi et al. [57].

4.2 Architecture

In order to train a model capable of learning dynamic gestures from a ToF camera, we adopted a CNNLSTM architecture, as has been done in similar [57] problems. The idea behind CNNLSTM architectures is as follows. A series of convolutional layers extract visual features from the images. Then, an LSTM layer extracts the time dependencies in these features. Finally, a series of fully connected layers are in charge of classifying the output of the LSTM layer into one of the six gesture types with a softmax activation function. Each convolutional layer consists of a convolution layer, a drop-out layer for regularization, and a max pooling layer. The output of the LSTM layer also goes through a dropout layer for regularization. The network was trained with a Tesla P100-SXM2-16GB GPU on simple gesture samples with a batch size of 48. The filter size for the first two convolutional layers was 5x5 and for the third convolutional layer was 3x3. Figure 2 contains more details about the architecture.

4.3 Baseline Training

Before adapting the models, it was first necessary to create a baseline: the universal background model (UBM). Training the network with the preprocessing steps described above resulted in the model quickly learning to classify the training gestures. Figure 3 shows that the network overfits roughly around epoch 30. However, testing the best model on the baseline evaluation set resulted in an accuracy of 66.9%. As can be seen in the confusion matrix in Figure 4, while some gestures, such as the pointing and two-finger gestures, are easy to recognize, differentiating between different directions of the rotate and swipe gestures can be more tricky. This could be due to the lack of data and individual differences in the participants in the universal background model and the participants chosen for the adaptation set. A way to address two of these problems was through adaptation.

4.4 Single-step Baseline Adaptation

While 66.9% accuracy may not be good enough to interact with a vehicle, we wanted to see if adding some gestures of particular users could improve the recognition rate of those particular users. During

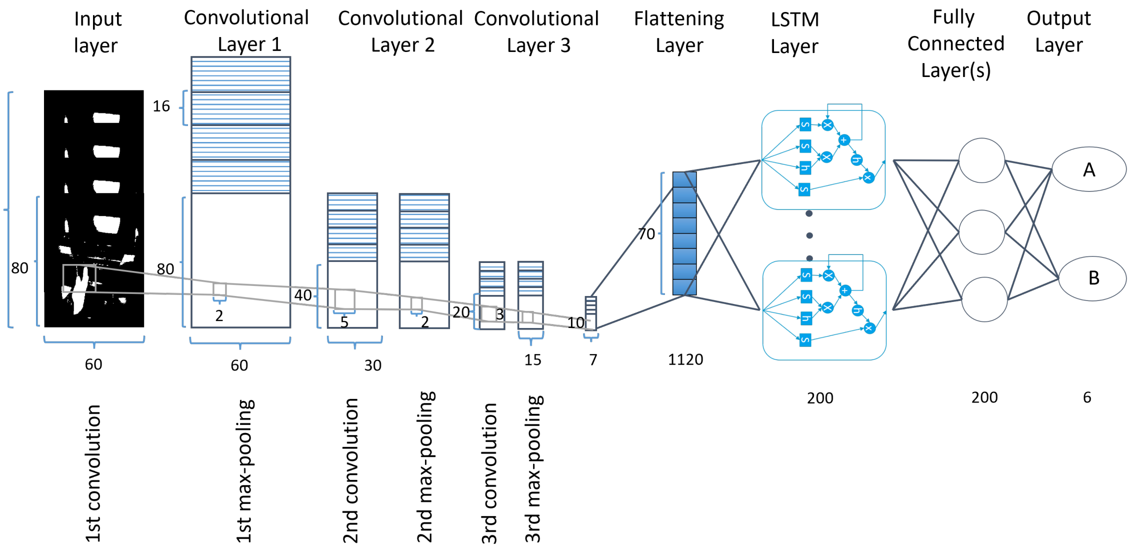


Figure 2: CNNLSTM architecture

the adaptation, the trained UBMs were subsequently trained (or fine-tuned) for the participants in the AS group. The UBM was trained for 20 additional epochs, with a batch size of 12 for each subject individually, using the training and validation data in the adaptive (AS) set. No layers were frozen during this additional training and the model was allowed to learn across all weights. This increased the accuracy of the evaluation data to 77.8%. As can be seen from the confusion matrix in Figure 4, the performance improved greatly, especially on the swipe and rotate gestures. However, as this could strongly depend on the subset of participants that make up the UBM and AS sets, we performed cross-validation, changing the participants of each set. This still resulted in a similar average improvement, with the UBM having 64% accuracy, compared to 70.7% after adaptation.

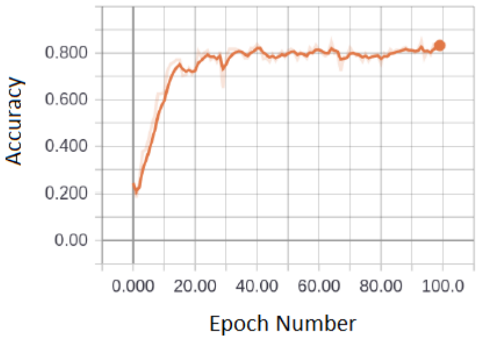


Figure 3: UBM model validation accuracy. The image shows the training progression in epochs. The model has reached its best performance around epoch 30 with a validation accuracy of about 80 percent.

4.5 Data Augmentation

While 70.7% in accuracy is undoubtedly an improvement over 64%, it is still lacking in human-computer interaction. Since DL models typically require vast amounts of data, a possible way to improve the model is by incorporating additional data into the training. However, getting new data can be expensive and time-consuming. For this reason, finding artificial ways to create more gestures could improve the model's performance. There are different ways to create these artificial data, such as using 3D software to model realistic gestures in a virtual environment, using generative adversarial neural networks, or modifying the collected real dataset itself. The latter is known as data augmentation. In our experiments, we tried a straightforward form of data augmentation consisting of adding random translations along the X and Y axes of -10 to +20 pixels. These simple frame modifications effectively augment the dataset artificially, helping the model's generalization, which resulted in the UBM improving to an average of 79.9% accuracy and the AS to an average of 86.4%.

4.6 Amount of Training Data for Adaptation

Given that both data augmentation and additional user-specific data benefit the model, an important question is how much data still benefit the model. For these experiments, we trained the models with the extended dataset (see Table 2). Adding gestures of more participants, together with the data augmentation method, has already increased the accuracy of the UBM to 84.7%. The next step was to train the AS models with different amounts of new gestures. We trained the AS models using 2, 8, 14, and 20 gestures per gesture class. The results were as expected: the more gestures the AS model was fine-tuned with, the better the performance (see Figure 5). The model's performance is already better with only two more gestures per class at 87.8% accuracy and peaks at around 14 to 20 gestures per class with 90.4% and 90.5% accuracy, respectively.

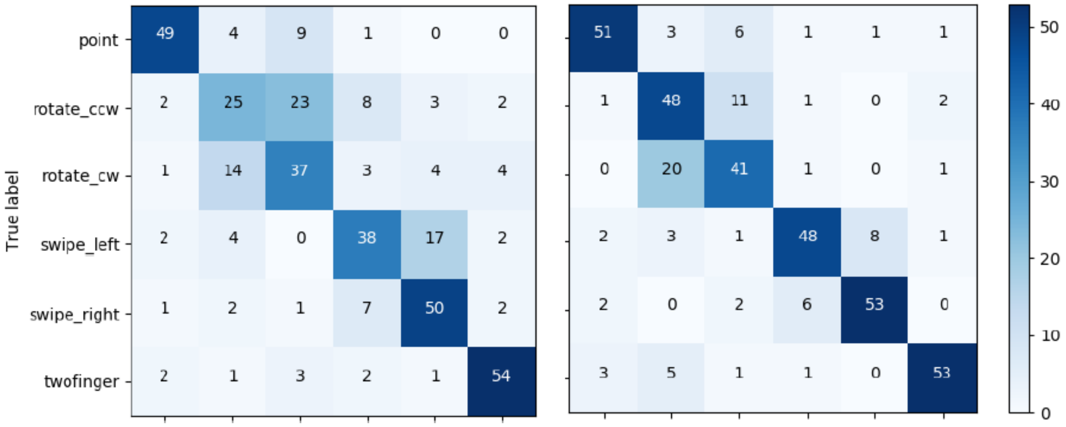


Figure 4: Left: Confusion matrix on baseline evaluation data for UBM (66.9% accuracy). Right: Confusion matrix on baseline evaluation data after single-step adaptation for AS (77.8% accuracy).

4.7 Incremental Learning

As seen in subsection 4.6, even small amounts of data can already improve the model. However, it is not always practical to collect the data beforehand. For example, a customer buying a car might be interested in having their personal gestures learned by the system. The customer could then make an appointment and record their gestures, repeating each of the gestures 20 times. They would then need to wait until the system is re-trained and updated with the new gesture recognition model. While this is feasible with a low number of gesture classes, the more tedious this procedure becomes as more gestures are incorporated into the system. Not only that, but every time the manufacturer introduces a new gesture type, the customer would have to return to the agency to repeat the process. A better solution would be if the vehicle itself could learn the user's gestures. This could be done by either the user manually recording and labeling the gestures or by the vehicle automatically labeling the user's gestures during regular interaction. However, how this system could be implemented is beyond the scope of this work. The question we address is how the model could be trained. A problem that might arise from this form of training is that the classifier needs to look at a wide distribution of the data in order to generalize well. If too little data are provided for training, this could have the risk of poor generalization. As one person only uses vehicles at a time, driver-specific models could better detect the gestures of particular drivers. However, if the model is overfitted to specific gestures done by the driver, slight variations in the gestures of the same driver could also be wrongly classified. In what follows, we perform some experiments on how such a model could be trained. In Figure 6 and Figure 7, the Y-axis represents the accuracy in the percentage of the evaluation set in the extended data set. The *dotted horizontal line* represents the performance of a single UBM from which the adapted models are incrementally trained. The *red line* represents the performance of an adapted model. These models were adapted to a single participant from the 11 participants that make up the adaptive set in the extended data set. The *blue line* shows the performance of the same model tested on the evaluation data of all other participants in the evaluation set (i.e., the remaining ten participants).

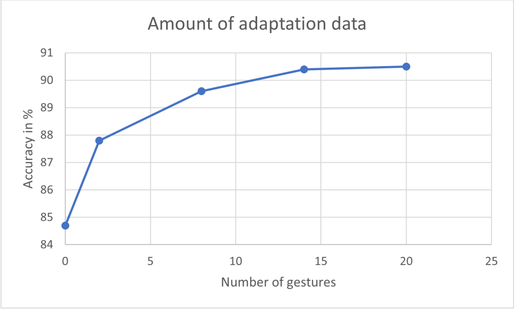


Figure 5: Performance of adapted models according to the number of gestures added to the AS model. The X-axis shows the number of gestures per class with which the model was trained.

4.7.1 Batch Size. In the first experiment, three update rates were compared, each corresponding to the batch size for the adaptation. That is, each data point represents an adaptation and also exactly one internal update of the network, using the same number of samples from each gesture class. Furthermore, for all three experiments, the same samples were used in the same order to ensure comparability. The evaluation set consisted of five individuals from the extended set whose results were averaged in this presentation. In particular, persons with less good initial values in the UBM were chosen in order to be able to observe the improvement better. For this reason, the red line in the cases shown also starts below the dashed UBM line. The starting point for adaptation represents the previous adapted network in each case. It was adapted with one single epoch in each batch, and no validation set was used. The lack of validation data was important because, in the real world, the availability of correctly labeled validation data for all persons is not guaranteed. Therefore, we wanted to experiment with the feasibility of adapting the model without validation data.

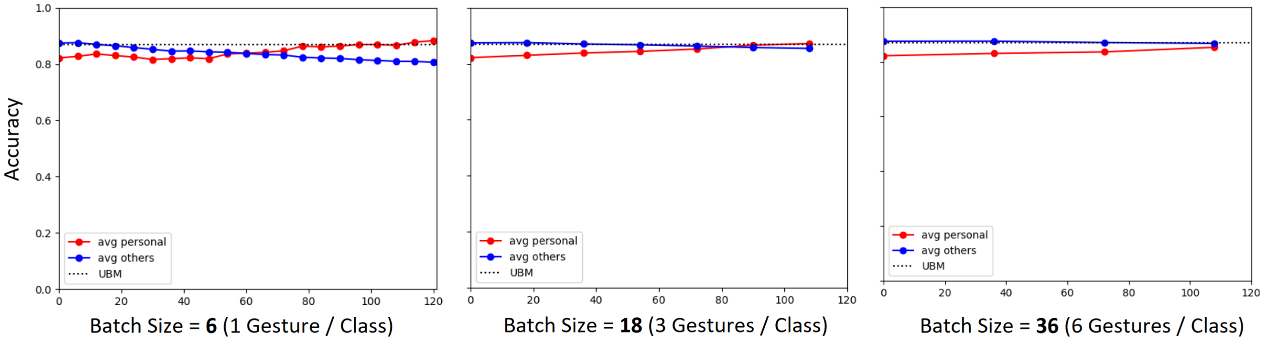


Figure 6: Incremental learning based on different update rates and batch sizes. On the left, one new gesture per class is used for the update. In the middle, three gestures per class. On the right, six gestures per class.

The following hypotheses were conceptualized for this approach and should be manifest in Figure 6:

- H1: The red line should (tend to) increase as the model improves for the adapted person by adding new gestures.
- H2: The blue line should (tend to) decrease because the adapted model now works less well for other persons. (This "forgetting" is a general side effect of personalization).
- H3: The accuracy at the right end of the X-axis should, in theory, be about the same regardless of the update rates for the red line.
- H4: The experiments with small batch size should show more significant fluctuations within the same interval on the X-axis than the graphs with larger batch size (Stability-Plasticity dilemma).

As seen in Figure 6, the red line is mainly increasing, albeit with fluctuations. The fluctuations are probably mainly due to the fact that no validation set was used, meaning that continuous improvement cannot be guaranteed. This is consistent with hypothesis H1. Additionally, the blue line decreases as expected (hypothesis H2), although this is mainly the case where a single gesture per class is used. This reflects the stability-plasticity dilemma: a reactive system that adapts quickly to changes in the data also tends to "forget" previous data faster. Updates with smaller batch sizes allow for a faster adaptation of the model but also fluctuate more strongly (hypothesis H4) Although the accuracy achieved at the end of the three settings deviates by up to 3% from each other, when the batch size 6 setting reaches gesture 108, it closely matches the other settings where they reach the same number.

It should also be noted that while these results reflect the average improvement, the degree to which the model was adapted for each individual was different. For some participants, the model improved continuously as new data was added. For some, the model performance first went down before going back up; for others, the recognition stayed mainly the same throughout. Therefore, a significant improvement through adaptation cannot be guaranteed in all cases. Especially with lower update rates, more substantial fluctuations in the model's performance are expected.

4.7.2 Sample Weight. In another experiment, we evaluated the influence of the sample weight as a means of correcting for learning speed and, thus, controlling the stability-plasticity of the model. We hypothesized that a lower sample weight should reduce the adaptation rate. This could counterbalance the fast adaptation rate for small batch sizes or the slow adaptation rate for large batch sizes. We compare two settings. In one, the samples are given a weight of 1, whereas, in the other, the samples are given a weight of 0.5. The results suggest that the higher sample weight indeed results in faster adaptation, and thus it can be used to speed up or slow down the adaptation of the model.

4.7.3 Use of Validation Data. This set of experiments investigates the influence of the update rate when validation data are available and multiple epochs can be performed during adaptation. The models were trained for 15 epochs, and of those, the overall best model is used as the base model for adaptation at each step. A separate set of samples of the adaptation subjects of 8 samples per class is used as the validation set. It was expected that these models would perform better than the previous models. However, since the best model selected is based only on the validation data, this was not guaranteed. With the above-mentioned parameters, deterioration between the data points on the X-axis can be almost completely avoided, i.e., the model continuously improves (see Figure 7). However, this behavior can only be achieved with a suitable validation set. In practice, however, this would imply that the driver's data would have to be recorded manually by the drivers themselves or in the agency. Additionally, using validation data appears to further improve the performance of the adapted models for the individuals while equally deteriorating the performance for other subjects. This could most likely have to do with the number of epochs for which the models were trained, as each epoch constitutes an update of the model. The decrease in accuracy for other participants is, however, not a problem in the case of intelligent vehicles, as these could store these personalized models for each of the drivers and switch them together with the driving profiles. It is also interesting that in the case of batch sizes 18 and 36, the increase and decrease in performance for the individual and the rest, respectively, seem more stable and pronounced. In the case where the batch size is six, the performance appears worse than when the validation set is not used for both the adapted individual and the rest of the participants.

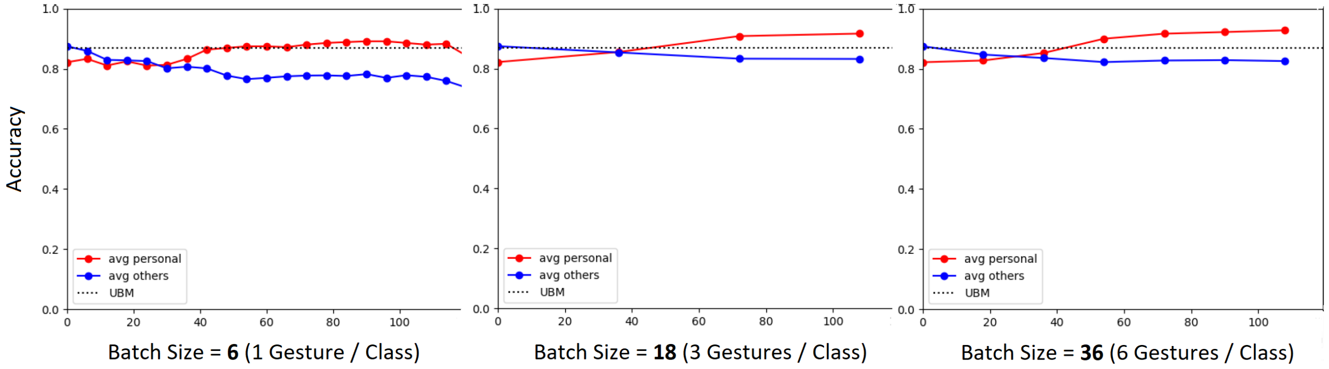


Figure 7: Progression of the incremental learning using validation data. Each data point along the line represents an update step of the network, although the network is updated throughout 15 epochs.

This, again, could have to do with the number of updates (epochs) to the network that are performed at each step.

5 CONCLUSION

This work demonstrates the feasibility of implementing dynamic gesture recognition in the automotive domain, even when data is limited. Although transfer learning can address the issue of limited data, its suitability may be domain-specific due to variations in data properties. To overcome this limitation, we collected a dataset of 4325 depth-based gestures, which was later extended to 7288. However, training a CNNLSTM model required additional data, prompting the development of preprocessing guidelines for time-of-flight sensors to mitigate the data scarcity problem. In addition, we propose data enhancement and incremental learning techniques to adapt the learning model and enhance the accuracy of a universal background model. The significance of these experiments and techniques in the context of in-vehicle human-machine interaction lies in their ability to facilitate model adaptation to the driver's interaction behavior. Furthermore, this study offers valuable insights and guidelines on effectively utilizing limited data to train dynamic gesture recognition algorithms with satisfactory performance, which can be extrapolated to other domains. While the internal validity is maximized in this work, future research will focus on examining the external validity of our model adaptation system in a driving environment, where additional challenges in data acquisition techniques arise.

ACKNOWLEDGMENTS

This work is partially funded by the German Ministry of Education and Research (BMBF) under the CAMELOT project (Grant Number: 01IW20008).

REFERENCES

[1] 2019. BMW's Innovative Gesture Control Technology Sets the Industry Standard. Accessed 12-01-2023. https://news.indigoautogroup.com/bmws-innovative-gesture-control-technology-sets-the-industry-standard/

[2] 2021. The new Mercedes-Maybach S-Class up close: MBUX Interior Assist Rear. Accessed 12-01-2023. https://group-media.mercedes-benz.com/marsMediaSite/en/instance/ko/The-new-Mercedes-Maybach-S-Class-up-close-MBUX-Interior-Assist-Rear.xhtml?oid=50185650

[3] Abdul Rafey Aftab, Michael von der Beeck, and Michael Feld. 2020. You Have a Point There: Object Selection Inside an Automobile Using Gaze, Head Pose and Finger Pointing. In *Proceedings of the 2020 International Conference on Multimodal Interaction*. Association for Computing Machinery, New York, NY, USA, 595–603. https://doi.org/10.1145/3382507.3418836

[4] Bashar I. Ahmad, Chrisminder Hare, Harpreet Singh, Arber Shabani, Briana Lindsay, Lee Skrypchuk, Patrick Langdon, and Simon Godsill. 2018. Selection facilitation schemes for predictive touch with mid-air pointing gestures in automotive displays. In *Proceedings of the 10th International Conference on Automotive User Interfaces and Interactive Vehicular Applications*. ACM, 21–32.

[5] Stephen R Anderson. 2010. How many languages are there in the world. *Linguistic Society of America* (2010). 6 pages.

[6] Joanna J. Bryson and Andreas Theodorou. 2019. *How Society Can Maintain Human-Centric Artificial Intelligence*. Springer Singapore, Singapore, 305–323.

[7] Nabil Al Nahin Ch, Diana Tosca, Tyanna Crump, Alberta Ansah, Andrew Kun, and Orit Shaer. 2022. Gesture and Voice Commands to Interact With AR Windshield Display in Automated Vehicle: A Remote Elicitation Study. In *Proceedings of the 14th International Conference on Automotive User Interfaces and Interactive Vehicular Applications* (Seoul, Republic of Korea) *(AutomotiveUI '22)*. Association for Computing Machinery, New York, NY, USA, 171–182. https://doi.org/10.1145/3543174.3545257

[8] Feng-Sheng Chen, Chih-Ming Fu, and Chung-Lin Huang. 2003. Hand gesture recognition using a real-time tracking method and hidden Markov models. *Image and vision computing* 21, 8 (2003), 745–758.

[9] Tuan Linh Dang, Sy Dat Tran, Thuy Hang Nguyen, Suntae Kim, and Nicolas Monet. 2022. An improved hand gesture recognition system using keypoints and hand bounding boxes. *Array* (2022), 100251. https://doi.org/10.1016/j.array.2022.100251

[10] Laura Dipietro, Angelo M Sabatini, and Paolo Dario. 2008. A survey of glove-based systems and their applications. *IEEE transactions on systems, man, and cybernetics, part c (applications and reviews)* 38, 4 (2008), 461–482.

[11] Stefan Eickeler, Andreas Kosmala, and Gerhard Rigoll. 1998. Hidden markov model based continuous online gesture recognition. In *Proceedings. Fourteenth International Conference on Pattern Recognition (Cat. No. 98EX170)*, Vol. 2. IEEE, 1206–1208.

[12] Hessam Jahani Fariman, Hasan J. Alyamani, Manolya Kavakli, and Len Hamey. 2016. Designing a user-defined gesture vocabulary for an in-vehicle climate control system. In *Proceedings of the 28th Australian Computer-Human Interaction Conference*. ACM, 391–395.

[13] Kikuo Fujimura, Lijie Xu, Cuong Tran, Rishabh Bhandari, and Victor Ng-Thow-Hing. 2013. Driver queries using wheel-constrained finger pointing and 3-D head-up display visual feedback. In *Proceedings of the 5th International Conference on Automotive User Interfaces and Interactive Vehicular Applications*. ACM, 56–62.

[14] Alexander Gepperth and Barbara Hammer. 2016. Incremental learning algorithms and applications. In *European symposium on artificial neural networks (ESANN)*.

[15] Amr Gomaa. 2022. Adaptive User-Centered Multimodal Interaction towards Reliable and Trusted Automotive Interfaces. In *Proceedings of the 2022 International Conference on Multimodal Interaction* (Bengaluru, India) *(ICMI '22)*. Association for Computing Machinery, New York, NY, USA, 690–695. https://doi.org/10.1145/3536221.3557034

[16] Amr Gomaa, Guillermo Reyes, Alexandra Alles, Lydia Rupp, and Michael Feld. 2020. Studying Person-Specific Pointing and Gaze Behavior for Multimodal Referencing of Outside Objects from a Moving Vehicle. In *Proceedings of the 2020 International Conference on Multimodal Interaction*. Association for Computing Machinery, New York, NY, USA, 501–509. https://doi.org/10.1145/3382507.3418817

[17] Amr Gomaa, Guillermo Reyes, and Michael Feld. 2021. ML-PersRef: A Machine Learning-Based Personalized Multimodal Fusion Approach for Referencing Outside Objects From a Moving Vehicle. In *Proceedings of the 2021 International Conference on Multimodal Interaction*. Association for Computing Machinery, New York, NY, USA, 318–327. https://doi.org/10.1145/3462244.3479910

[18] Aashni Haria, Archanasri Subramanian, Nivedhitha Asokkumar, Shristi Poddar, and Jyothi S. Nayak. 2017. Hand gesture recognition for human computer interaction. *Procedia Computer Science* 115 (2017), 367–374.

[19] Deng-Yuan Huang, Wu-Chih Hu, and Sung-Hsiang Chang. 2011. Gabor filter-based hand-pose angle estimation for hand gesture recognition under varying illumination. *Expert Systems with Applications* 38, 5 (2011), 6031–6042.

[20] Pan Jing and Guan Ye-Peng. 2013. Human-computer interaction using pointing gesture based on an adaptive virtual touch screen. *International Journal of Signal Processing, Image Processing and Pattern Recognition* 6, 4 (2013), 81–91.

[21] Roland Kehl and Luc Van Gool. 2004. Real-time pointing gesture recognition for an immersive environment. In *Proceedings of the 6th International Conference on Automatic Face and Gesture Recognition*. IEEE, 577–582.

[22] Okan Köpüklü, Ahmet Gunduz, Neslihan Kose, and Gerhard Rigoll. 2019. Real-time hand gesture detection and classification using convolutional neural networks. In *2019 14th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2019)*. IEEE, 1–8.

[23] Bilawal Latif, Neil Buckley, and Emanuele Lindo Secco. 2023. Hand Gesture and Human-Drone Interaction. In *Intelligent Systems and Applications*, Kohei Arai (Ed.). Springer International Publishing, Cham, 299–308.

[24] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep learning. *nature* 521, 7553 (2015), 436–444.

[25] Gongfa Li, Heng Tang, Ying Sun, Jianyi Kong, Guozhang Jiang, Du Jiang, Bo Tao, Shuang Xu, and Honghai Liu. 2019. Hand gesture recognition based on convolution neural network. *Cluster Computing* 22, 2 (2019), 2719–2729.

[26] Hsien-I Lin, Ming-Hsiang Hsu, and Wei-Kai Chen. 2014. Human hand gesture recognition using a convolution neural network. In *2014 IEEE International Conference on Automation Science and Engineering (CASE)*. IEEE, 1038–1043.

[27] Garg Mallika, Debashis Ghosh, and Pyari Mohan Pradhan. 2023. A Two-Stage Convolutional Neural Network for Hand Gesture Recognition. In *Proceedings of the 6th International Conference on Advance Computing and Intelligent Engineering*, Bibudhendu Pati, Chhabi Rani Panigrahi, Prasant Mohapatra, and Kuan-Ching Li (Eds.). Springer Nature Singapore, Singapore, 383–392.

[28] Manar Maraqa and Raed Abu-Zaiter. 2008. Recognition of Arabic Sign Language (ArSL) using recurrent neural networks. In *2008 First International Conference on the Applications of Digital Information and Web Technologies (ICADIWT)*. IEEE, 478–481.

[29] Tin Hninn Hninn Maung. 2009. Real-time hand tracking and gesture recognition system using neural networks. *International Journal of Computer and Information Engineering* 3, 2 (2009), 315–319.

[30] Byung-Woo Min, Ho-Sub Yoon, Jung Soh, Yun-Mo Yang, and Toshiaki Ejima. 1997. Hand gesture recognition using hidden Markov models. In *1997 IEEE International Conference on Systems, Man, and Cybernetics. Computational Cybernetics and Simulation*, Vol. 5. IEEE, 4232–4235.

[31] Pavlo Molchanov, Shalini Gupta, Kihwan Kim, and Jan Kautz. 2015. Hand gesture recognition with 3D convolutional neural networks. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*. IEEE, 1–7.

[32] Pavlo Molchanov, Shalini Gupta, Kihwan Kim, and Kari Pulli. 2015. Multi-sensor system for driver's hand-gesture recognition. In *Proceedings of the 11th International Conference on Automatic Face and Gesture Recognition*. IEEE, 1–8.

[33] Mohammad Mehdi Moniri and Christian Müller. 2012. Multimodal reference resolution for mobile spatial interaction in urban environments. In *Proceedings of the 4th International Conference on Automotive User Interfaces and Interactive Vehicular Applications*. ACM, 241–248.

[34] Jawad Nagi, Frederick Ducatelle, Gianni A Di Caro, Dan Cireşan, Ueli Meier, Alessandro Giusti, Farrukh Nagi, Jürgen Schmidhuber, and Luca Maria Gambardella. 2011. Max-pooling convolutional neural networks for vision-based hand gesture recognition. In *2011 IEEE international conference on signal and image processing applications (ICSIPA)*. IEEE, 342–347.

[35] Robert Neßelrath, Mohammad Mehdi Moniri, and Michael Feld. 2016. Combining speech, gaze, and micro-gestures for the multimodal control of in-car functions. In *Proceedings of the 12th International Conference on Intelligent Environments*. IEEE, 190–193.

[36] Natalia Neverova, Christian Wolf, Giulio Paci, Giacomo Sommavilla, Graham Taylor, and Florian Nebout. 2013. A multi-scale approach to gesture detection and recognition. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*. 484–491.

[37] Kai Nickel, Edgar Scemann, and Rainer Stiefelhagen. 2004. 3D-tracking of head and hands for pointing gesture recognition in a human-robot interaction scenario. In *Proceedings of the 6th International Conference on Automatic Face and Gesture Recognition*. IEEE, 565–570.

[38] Andrzej Nowak, Paul Lukowicz, and Pawel Horodecki. 2018. Assessing artificial intelligence for humanity: Will AI be the our biggest ever advance? Or the biggest threat [Opinion]. *IEEE Technology and Society Magazine* 37, 4 (2018), 26–34.

[39] Eshed Ohn-Bar and Mohan Manubhai Trivedi. 2014. Hand gesture recognition in real time for automotive interfaces: A multimodal vision-based approach and evaluations. *IEEE Transactions on Intelligent Transportation Systems* 15, 6 (2014), 2368–2377.

[40] Aye Su Phyo, Hisato Fukuda, Antony Lam, Yoshinori Kobayashi, and Yoshinori Kuno. 2019. A Human-Robot Interaction System Based on Calling Hand Gestures. In *Intelligent Computing Methodologies*, De-Shuang Huang, Zhi-Kai Huang, and Abir Hussain (Eds.). Springer International Publishing, Cham, 43–52.

[41] Carl A Pickering, Keith J Burnham, and Michael J Richardson. 2007. A research study of hand gesture recognition technologies and applications for human vehicle interaction. In *Proceedings of the 3rd Institution of Engineering and Technology conference on automotive electronics*. IET, 1–15.

[42] Raimundo F Pinto, Carlos DB Borges, Antônio Almeida, and Iális C Paula. 2019. Static hand gesture recognition based on convolutional neural networks. *Journal of Electrical and Computer Engineering* 2019 (2019).

[43] David Rempel, Matt J. Camilleri, and David L. Lee. 2014. The design of hand gestures for human-computer interaction: Lessons from sign language interpreters. *International Journal of Human Computer Studies* 72, 10-11 (10 2014), 728–735.

[44] Yu Ren and Fengming Zhang. 2009. Hand gesture recognition based on MEB-SVM. In *2009 International Conference on Embedded Software and Systems*. IEEE, 344–349.

[45] Gerhard Rigoll, Andreas Kosmala, and Stefan Eickeler. 1997. High performance real-time gesture recognition using hidden markov models. In *International Gesture Workshop*. Springer, 69–80.

[46] Florian Roider and Tom Gross. 2018. I see your point: Integrating gaze to enhance pointing gesture accuracy while driving. In *Proceedings of the 10th International Conference on Automotive User Interfaces and Interactive Vehicular Applications*. ACM, 351–358.

[47] Florian Roider, Sonja Rümelin, Bastian Pfleging, and Tom Gross. 2017. The effects of situational demands on gaze, speech and gesture input in the vehicle. In *Proceedings of the 9th International Conference on Automotive User Interfaces and Interactive Vehicular Applications*. ACM, 94–102.

[48] Sonja Rümelin, Chadly Marouane, and Andreas Butz. 2013. Free-hand pointing for identification and interaction with distant objects. In *Proceedings of the 5th International Conference on Automotive User Interfaces and Interactive Vehicular Applications*. ACM, 40–47.

[49] Tara N Sainath, Oriol Vinyals, Andrew Senior, and Haşim Sak. 2015. Convolutional, long short-term memory, fully connected deep neural networks. In *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 4580–4584.

[50] Ben Shneiderman. 2020. Human-centered artificial intelligence: Reliable, safe & trustworthy. *International Journal of Human–Computer Interaction* 36, 6 (2020), 495–504.

[51] Dushyant Kumar Singh. 2015. Recognizing hand gestures for human computer interaction. In *Proceedings of the International Conference on Communications and Signal Processing*. IEEE, 379–382.

[52] Ekaterini Stergiopoulou and Nikos Papamarkos. 2009. Hand gesture recognition using a neural network shape fitting technique. *Engineering Applications of Artificial Intelligence* 22, 8 (2009), 1141–1158.

[53] Dina Stiegemeier, Sabrina Bringeland, Johannes Kraus, and Martin Baumann. 2022. User Experience of In-Vehicle Gesture Interaction: Exploring the Effect of Autonomy and Competence in a Mock-Up Experiment. In *Proceedings of the 14th International Conference on Automotive User Interfaces and Interactive Vehicular Applications* (Seoul, Republic of Korea) *(AutomotiveUI '22)*. Association for Computing Machinery, New York, NY, USA, 285–296. https://doi.org/10.1145/3543174.3546847

[54] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. 2015. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 1–9.

[55] Tomoichi Takahashi and Fumio Kishino. 1991. Hand Gesture Coding Based on Experiments Using a Hand Gesture Interface Device. *SIGCHI Bull.* 23, 2 (mar 1991), 67–74. https://doi.org/10.1145/122488.122499

[56] Naohiro Takemura, Toshio Inui, and Takao Fukui. 2018. A neural network model for development of reaching and pointing based on the interaction of forward and inverse transformations. *Developmental science* 21, 3 (2018). 10 pages.

[57] Eleni Tsironi, Pablo VA Barros, and Stefan Wermter. 2016. Gesture Recognition with a Convolutional Long Short-Term Memory Recurrent Neural Network.. In *ESANN*.

[58] Wei Xu. 2019. Toward human-centered AI: a perspective from human-computer interaction. *interactions* 26, 4 (2019), 42–46.

[59] Qi Ye, Lanqing Yang, and Guangtao Xue. 2018. Hand-free gesture recognition for vehicle infotainment system control. In *Proceedings of the IEEE Vehicular Networking Conference*. IEEE, 1–2.

[60] Dan Zhao, Cong Wang, Yue Liu, and Tong Liu. 2019. Implementation and evaluation of touch and gesture interaction modalities for in-vehicle infotainment systems. In *Image and Graphics*, Yao Zhao, Nick Barnes, Baoquan Chen, Rüdiger Westermann, Xiangwei Kong, and Chunyu Lin (Eds.). Springer, 384–394.

[61] Thomas G. Zimmerman, Jaron Lanier, Chuck Blanchard, Steve Bryson, and Young Harvill. 1986. A Hand Gesture Interface Device. In *Proceedings of the SIGCHI/GI Conference on Human Factors in Computing Systems and Graphics Interface* (Toronto, Ontario, Canada) *(CHI '87)*. Association for Computing Machinery, New York, NY, USA, 189–192. https://doi.org/10.1145/29933.275628

[62] Martin Zobl, Michael Geiger, Björn Schuller, Manfred Lang, and Gerhard Rigoll. 2003. A real-time system for hand gesture controlled operation of in-car devices. In *Proceedings of the International Conference on Multimedia and Expo (ICME)*. IEEE. 4 pages.