Free-hand Gesture Recognition with 3D-CNNs for In-car Infotainment Control in Real-time

Fabian Sachara
Computer Science Institute
Hochschule Ruhr West
46236 Bottrop, Germany
fabian.sachara@hs-rw.de

Thomas Kopinski
Fachhochschule Südwestfalen
Lindenstraße 52
59872 Meschede, Germany
kopinski.thomas@fh-swf.de

Alexander Gepperth
Applied Computer Science
University of Applied Sciences
36037 Fulda , Germany
alexander.gepperth@cs.hs-fulda.de

Uwe Handmann
Computer Science Institute
Hochschule Ruhr West
46236 Bottrop, Germany
uwe.handmann@hs-rw.de

***Abstract*—In this contribution we present a novel approach to transform data from time-of-flight (ToF) sensors to be interpretable by Convolutional Neural Networks (CNNs). As ToF data tends to be overly noisy depending on various factors such as illumination, reflection coefficient and distance, the need for a robust algorithmic approach becomes evident. By spanning a three-dimensional grid of fixed size around each point cloud we are able to transform three-dimensional input to become processable by CNNs. This simple and effective neighborhood-preserving methodology demonstrates that CNNs are indeed able to extract the relevant information and learn a set of filters, enabling them to differentiate a complex set of ten different gestures obtained from 20 different individuals and containing 600.000 samples overall. Our 20-fold cross-validation shows the generalization performance of the network, achieving an accuracy of up to 98.5% on validation sets comprising 20.000 data samples. The real-time applicability of our system is demonstrated via an interactive validation on an infotainment system running with up to 40fps on an iPad in the vehicle interior.**

I. Introduction

Free-hand gestures are an established means of control for various kinds of systems with a broad range of applications, however they are rarely used as a solitary interaction technique due to various limitations. There are different ways of approaching this task which can be distinguished between the immersive (using devices attached to your body) or non-immersive (using devices for observing the scene) approaches. Depending on the scenario, the most efficient way of providing this possibility has to be chosen while it remains clear that in an automotive environment the desirable way is to keep the degrees of freedom of the driver at a maximum. When also taking into account the challenging lighting conditions (day-night shift, direct sunlight) the means of observing the scene has to be carefully selected. We present a novel method for recognizing a challenging set of ten different hand gestures from time-of-flight (ToF) data recorded with the Creative Gesture Camera. Based on a large-scale hand gesture set recorded from 20 different individuals we train and optimize a Convolutional Neural Network to be able to distinguish the different gestures. CNNs are optimized to work on 2D data of fixed size while the resulting point clouds stemming from a ToF camera are voxel data of variable size - hence we propose a novel approach of transforming the data in order to become interpretable by CNNs. We evaluate the performance of our system via a series of leave-one-out cross-validation tests demonstrating its generalization capability on unknown persons. Lastly, we show the applicability of the system by setting up a demonstrator utilizing the object recognition of our system on a mobile tablet with an infotainment application specifically tailored to the task. The rest of this contribution is laid out as follows: The following section presents the most important related work to the topic. Section III shows the core of the data transformation step and the resulting network architecture. The backbone of our system is a large-scale database which is presented and explained in Section IV. To validate the presented approach we conducted a series of experiments which are described in Section V. We discuss a possible system setup with an infotainment application specifically developed to demonstrate its applicability in Section VI. Finally we conclude with a discussion of the most significant insights and give an outlook on future work in Section VII.

II. RELATED WORK

The is an abundant amount of work related to the topic of Hand Gesture Recognition (HGR) [1], [2], [3], [4], [5], [6], [7], [8], [9], [10], [11]. Within the field of Advanced Driver Assistance Systems some of the main challenges remain the illumination interferences, real-time capability, scaling, rotation and translation as well as HMI-related issues (gesture set, interaction area etc.). Deep Learning has been successfully applied to solve a plethora of Computer Vision problems, within the area of 3D Vision and more specifically HGR the amount of related work is yet scarce. Glatt [12] has shown how Deep Learning can be successfully applied to achieve HGR from Kinect data with the help of Deep Belief Networks. The best recognition results oscillate between 75% to 85% and are partially comparable scores achieved in this contribution although similar accuracy scores as high as >98% are never reached. Barros et al. [13] show how CNNs can effectively be applied to recognize Italian sign gestures from Kinect data achieving error scores of 8.3% for the best model with their system also working in real-time. Tang et al. [14] show how Deep Neural Networks can be utilized to discriminate between 20 different hand poses using a Kinect sensor achieving high accuracy ratings. Although the authors claim that illumination invariance is achieved while making use of both depth and

978-1-5386-1526-3/17/$31.00 ©2017 IEEE

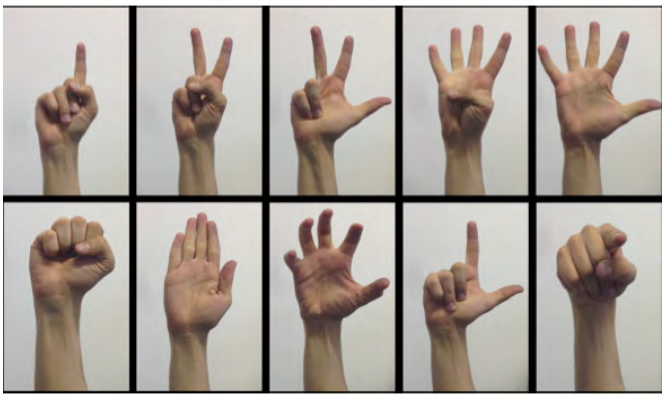

Fig. 1. The hand gesture database consisting of ten different gestures: *ONE*, *TWO*, *THREE*, *FOUR*, *FIVE*, *FIST*, *FLAT*, *GRAB*, *PINCH*, *POINT*

color data, how exactly this goal is reached remains unclear as both RGB and depth measurements stemming from a Kinect sensor are unreliable when exposed to direct sunlight. Given these circumstances, the approach presented in this contribution supposes relying on a single depth sensor to achieve illumination invariance and real-time capability as modern ToF cameras are able to record data with up to 90fps. The novelty in this approach lies in the light-weight data transformation step which not only allows for CNNs to be utilized but moreover maintains real-time performance. This combination of low-cost hardware and rapid algorithmic processing yields high recognition results and is, to the best of our knowledge, a novel contribution to the field of HGR for infotainment control with CNNs.

III. Data preparation and Network structure

In order to be able to deal with three-dimensional input, this contribution presents an approach which transforms the raw 3D data into a format readable by CNNs. The need for a fixed-size input requires a specific partitioning of the 3D input. Given an input of 3D data points (voxels) of arbitrary extension across all possible dimensions (also referred to as point cloud), we propose the subdivision of the entire cloud into cubes of fixed size. To this end, the maximal extension of the data points has to be calculated for the entire problem. The underlying approach utilizes the 3D-subdivision of the point cloud with the subsequent summation of the points within each resulting cube.

Raw data coming from depth sensors describes the environment in a 2.5 dimensional way. As opposed to e.g. 3D models of objects created by hand, a sensor has limited view onto a scene namely in that the vision is bound by the sensor's perspective, subsequently a lot of information is irretrievable because it is hidden from the observer. This presents an obstacle to our approach as we try to create input data readable for CNNs from a reduced set of data points. To this end, in order to be able to work on 3D input data we employ a modified LeNet 5 implementation of the Theano library [15] [16] with two convolutional layers. The input space is subdivided into n^3 hypercubes of fixed size. Each hypercube then contains a subset of data points from the original object. Depending on the density of the cloud, a certain number of cubes remains empty. In order to avoid too many empty hypercubes, which form the input for the CNN, we stretch the data to fit into the raster. To this end, the input cloud is normalized to the range (0,1) on each axis. This guarantees the data to be evenly distributed over all hypercubes. The value contained within a hypercube is determined by the number of data points it contains [17].

Each slice of the input vector, which will be described here on basis of an $8 \times 8 \times 8$ sized example, has to be reshaped to fit a designated pattern: The vector is reshaped in a way that each row fed into the convolutional layer represents one (x-y) slice of depth data in the original, resulting in an input matrix of 8×64 (cf. Figure 2 showing this for the case of 4^3). This way, a convolutional kernel of size 8×1 can be used to initially convolve the depth-axis, resulting in an 1×64 output of the first kernel. No max-pooling is used in this layer. The second layer reshapes this 1×64 output to 8×8, so that a 3×3 kernel can subsequently be utilized. This layer also implements 2×2 max-pooling, resulting in an output of 3×3. This output is then fed into the MLP layer of the convolutional net, which determines the output class. The resulting kernels obtained from the training run are depicted in Figure 3.

IV. The Database

To only capture the relevant data points which are part of the user's right hand, distance thresholding is introduced during the recording. Points recorded by the sensor are simply cropped if above a certain threshold value Θ. Furthermore, the recording takes place in a predefined Volume of Interest (VOI) to ignore irrelevant data points to the sides of the user's hand. The resulting data is denoted a point cloud (PC) of a posture. Our database comprises 10 different static hand postures recorded with the Creative Gesture Camera. The individual postures are denoted *ONE*, *TWO*, *THREE*, *FOUR*, *FIVE*, *FIST*, *FLAT*, *GRAB*, *PINCH*, *POINT* and described in Figure 1. These hand postures were chosen as a trade-off between meaningfulness and difficulty (in terms of disambiguation). With respect to the meaning of the postures, all of them can be facilitated to represent typical functions addressable in infotainment systems. Counting from one to five, for instance, can be used to switch radio channels. Other postures can be used to interact with audio (*FLAT*) or to choose elements (*POINT*). The difficulty in disambiguation results from the fact that the difference between some postures is defined by one finger only (e.g., *ONE* vs *TWO*) which, depending on the distance to the sensors, is equivalent to as few as 20-40 voxels.

The sensor is mounted in front of the user recording the nearby environment from an orthogonal angle. Each posture is performed and recorded 3000 times. In order to induce some variance into the data, each participant is asked to translate and rotate her/his hand during the recording. Furthermore, the

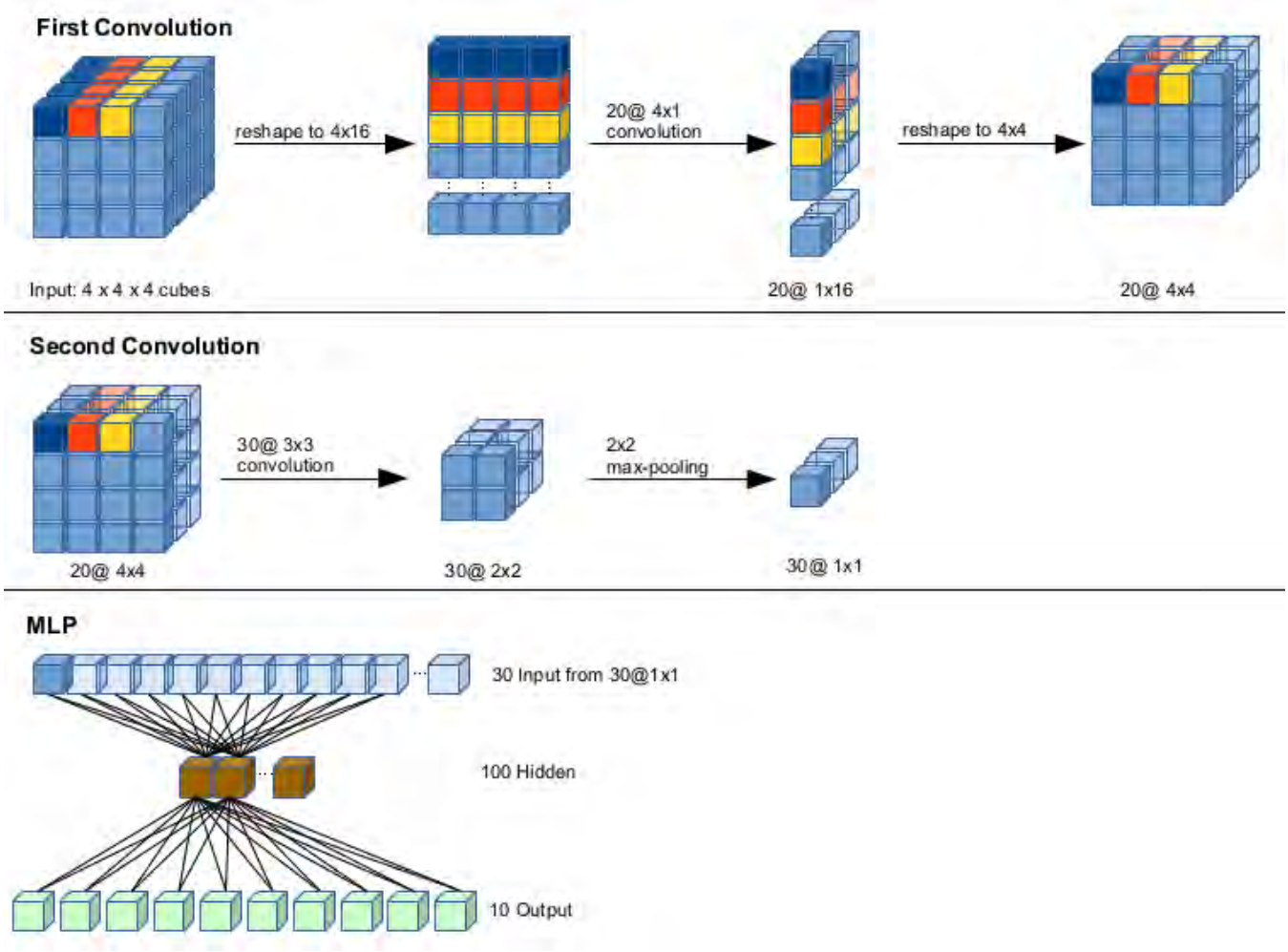

Fig. 2. Setup of the CNN structure with two convolutional layers. Top row: First convolution step and reshaping. Center: Second convolution step and max-pooling. Bottom: MLP structure and input.

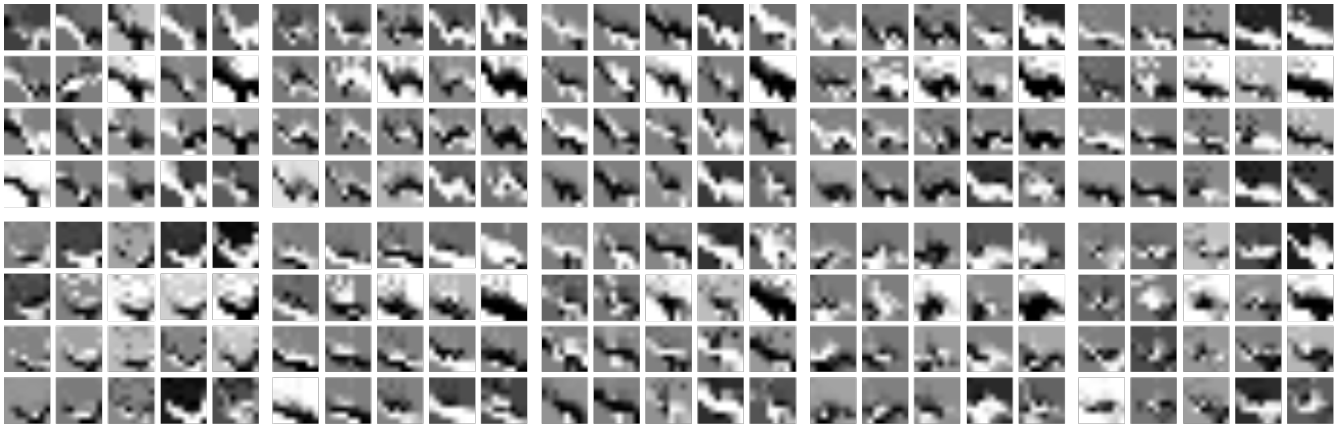

Fig. 3. The resulting kernels from the first filter grouped together for each posture from the data set (cf. Figure 1). The first layer of the CNN produces 20 different kernels. All 20 kernels produced per gesture are grouped and presented in analogous order from left to right, top to bottom.

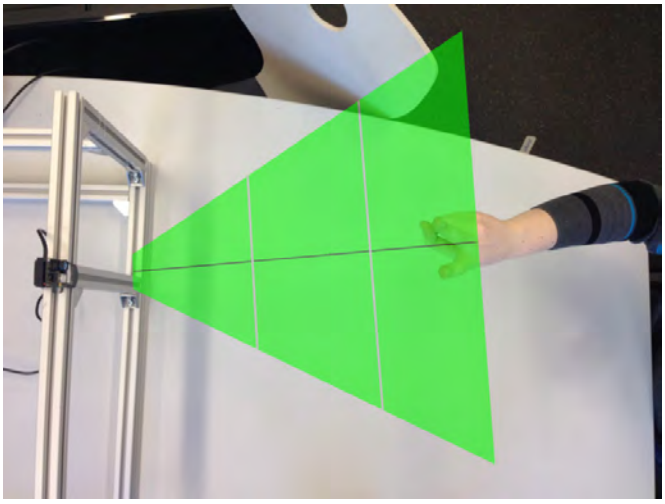

Fig. 4. The three zones of the recording: Near, intermediate and far.

recording area is divided into three zones: near (15-30cm), intermediate (30-45cm) and far (45-60cm) with respect to the distance between sensor and hand. During the recording the median of the captured PC is calculated and the sample is included into the database depending on the zone it is captured in. After 1000 samples have been captured within the near zone, the participant is asked to perform the next 1000 samples in the following zone. This way we ensure that the sum of recorded hand postures per participant is equally recorded over all three range zones.

The result of such a recording can be seen in Figure 5. The resulting point cloud is depicted for two different snapshots in subsequent movements (top vs. bottom) of the same participant from two different angles (left vs. right). Points closer to the sensor are depicted in yellow color, points further distant in a dark green color. Depending on the angle the user postures her/his hand toward the sensor, more or less light is reflected back and hence the precision of the measurement suffers. Another possible source for noise is the fact that depth measurement relies on the amount of light reflected from the object, however too much light reflected over-saturates the measurement. This is visible by the amount of noise (or outliers) existent in the image. In the upper row, the user poses in a rather orthogonal angle towards the sensor, therefore there are less outliers visible towards the edges of the object. As compared to the bottom row, more outliers are recognizable as can be seen in the front view (left) and the side view (right) of the same posture. Dealing with noise is an important factor for the task of hand posture recognition in particular as depth sensors typically have a lower resolution than RGB cameras and therefore data samples suffering from much noise tend to strongly impede the employed algorithms. Consequently, no filtering or noise reduction techniques have been utilized to remove said outliers. However, due to the movement performed by each individual during the recording, the amount of data points belonging to the forearm differs strongly as can be seen in

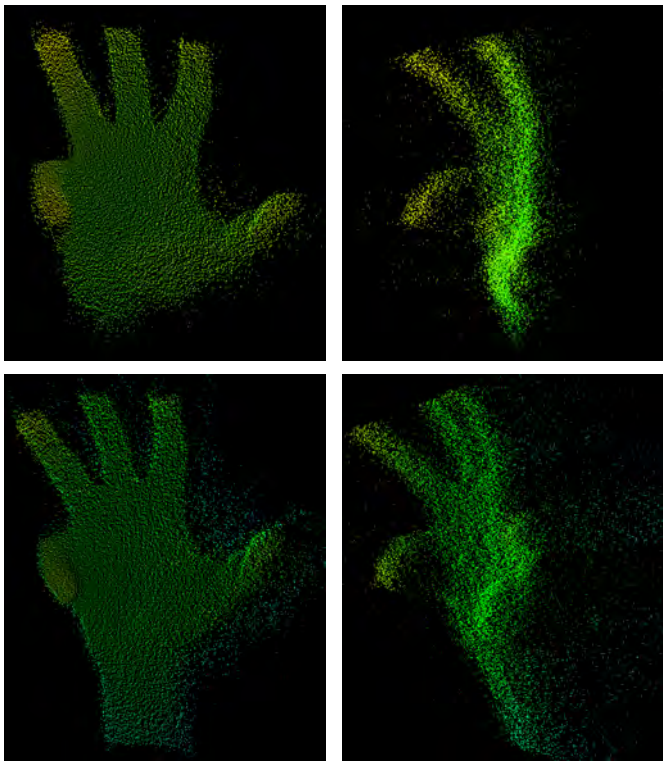

Fig. 5. Sample recording of a hand posture. Top row: The same posture from the front view (left) and the side view (right). Bottom row: The same hand posture in a subsequent state taken after the snapshot in the top row (same angles). Noise and outliers resulting from errors in measurement are clearly visible when seen from the side view (right column).

Figure 5 (top left vs. bottom left). Data points belonging to the forearm carry no information necessary to distinguish any of the posture in the database therefore we employed a cropping algorithm relying on a Principal Components Analysis (PCA) of the hand-arm object. Automatically removing most of the forearm results in a smaller first principal component, and more relevant information included in each sample, leaving only the palm and the fingers.

Results are visible in Figure 5 (top left vs. bottom left). As compared to the uncropped Point Cloud, the result of removing most of the forearm clearly shows the advantage of PCA cropping, namely the reduction to the essential parts. The following chapters demonstrate a sample infotainment application as well as the experiments and results of a standard pattern recognition algorithm. This is included as a performance baseline and to establish a well-defined experimental procedure, in order to allow other algorithms to be compared meaningfully.

V. Experiments and Results

The results of our experiment run are presented in Table I. We have conducted a series of experiments to test the validity of our approach on unseen data. To this end, each column represents the Classification Error (CE) achieved on Person p from the database when the network is trained on all the other

data samples except on data coming from person p. Thus Table I shows the results of an n-fold cross-validation run on the presented database. Experiments are conducted on an NVIDIA GTX 780 Ti using the Theano implementation of a CNN. As memory is limited for preparing and storing the data samples only 2000/3000 samples per gesture and person are randomly selected and taken for training and validation. Consequently, each run trains a CNN on 19 persons with 20.000 gestures samples each yielding a training set of 380.000 samples and a validation set of 20.000 samples.

The results show slightly varying, however very satisfactory performance. With the approach presented in this contribution we are able to demonstrate the robustness of the underlying methodology. Performance peaks with the model obtained for person 13 (1.5% error) and for 14 out of 20 persons error rates around or far below the 15% mark are achieved which is significant for such a large and diverse validation set of unseen data. This underlines the fact that CNNs paired with our data transformation technique are able to generalize well on this complex problem. Considering the fact that we are able to produce up to 40 classification steps per second this is more than enough to realize a real-time application which produces satisfactory behaviour. The main drawback is the somewhat poor performance on persons 2, 4, 7 and 19 with CEs ranging from 30,5% to 48,3% and has to be attributed to various factors such as e.g. user behaviour or noise in the measurements. An in-depth evaluation shows the very specific way in which the persons in question pose one and the same gesture during the process of the recording with a rapid change in finger positioning. This is one of the main reasons for the system's fluctuation in performance as the CNN tries to capture all the possible variants of the given classes which leads to problems in those cases where gestures are similar and somehow posed 'wrongly' by some participants. As for the training time, most of the CNNs converged quickly with as few as a thousand iterations (corresponding to roughly 1h train time) required for finding their global optimum in most cases with only one case (more than 22K iterations for person 14) requiring extensive parameter search. This is, however, the case which simultaneously corresponds to the model generalizing best with a CE of 1,5%. The termination condition for training is the non-improvement of the CE on the validation set for 100 subsequent epochs which in this case demonstrates the potential of extended optimization search. Execution time for a single sample falls well below 1 ms which is negligible in regarding system workload.

VI. System Setup and Live Demonstration

Our system consists of the Creative Gesture Camera recording the VOI in the interior of the vehicle just before the front console with a lateral resolution of 320×160 of the depth sensor [18]. The iPad is mounted to the front console and runs an application with typical infotainment scenarios (media, maps, contacts, phone, climate). Figure 7 shows four of the sixteen (sub-)screens of the infotainment system. Overall, typical functions such as media or navigation selection, navigating through submenus, browsing and turning music on/off are addressable through the freehand gestures.

Fig. 6. Our system setup with an iPad and the Creative Gesture Camera as described in Section VI.

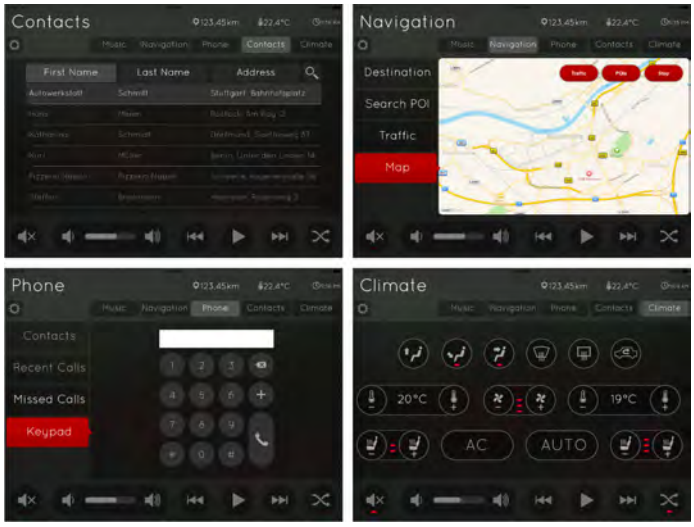

Fig. 7. Sample screens of the infotainment system running on a mobile tablet: Contacts, Navigation, Phone and Climate (top to bottom, left to right).

The camera is connected to a standard laptop which in turn is responsible for recording the VOI in the nearby driver zone, cropping of the recorded point cloud (PCA cropping to remove irrelevant arm parts) and processing the cloud to compute the features to subsequently pass them to the CNN for classification. We implemented an averaging window to record 20 snapshots in a row and produce a classification results for each. The final decision, which corresponds to the interpreted gesture sent to the iPad, is produced by max-voting. As our systems works with 35-40 fps this is more than sufficient to balance the confidence in decision-making and real-time capability. An additional delay after each gesture is sent, creates a more realistic setting for the application to be tested as the system pauses for interaction and proceeds with the received input[1].

[1]See the attached video for a live demonstration. Results in real-time are difficult to measure, however this short demonstration shows the robustness of our approach, its applicability and extendability as well as the work-flow.

TABLE I
CE PER PERSON (IN PERCENT, 2ND ROW) AND NUMBER OF ITERATIONS (IN THOUSAND, 3RD ROW) PER TRAINING RUN

Pers.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
CE	15.3	48.3	21.3	33.3	12.6	16.4	40.3	20.8	4.6	9.5	7.2	5.4	1.5	2.7	3.5	6.2	8.0	12.0	30.5	14.4
Iter.	8.3	1.1	3.7	1.5	1.2	0.6	1.8	5.5	11.9	1.9	7.9	9.1	1.0	22.2	3.6	6.5	2.0	4.5	1.7	1.6

VII. CONCLUSION AND OUTLOOK

This contribution shows the effective utilization of Deep Learning in form of a CNN for recognizing freehand gestures from depth data. We present a novel approach of transforming point clouds into a data format of fixed size allowing the CNN to extract the necessary information in form of filters to be learned for the task of object recognition. Our system is easy to set up as it requires only a single depth sensor to be positioned in roughly the same distance as during the recording of the database, thus avoiding the need for cumbersome calibration procedures. Once set up, the gesture recognition pipeline simply crops the vehicle interior, extracting depth information of arm, palm and finger position which is subsequently cropped again as to get rid of the 'irrelevant' part of the arm. The features are then extracted and presented to the CNN for classification. The neuron with the highest activation in the output layer corresponds to the class in question. Through a max-voting scheme we are able to take into account 20 consecutive snapshots for the decision making in order to stabilize the system over time. Not only is this a light-weight approach able to achieve up to 40 fps but it also allows to stabilize the system's performance achieving recognition rates of up to 100% during execution time for some gestures. The problem of scaling, translation and rotation is addressed through incorporation of many varying data samples during the recording of the database. The hand gesture recognition pipeline presented in this way demonstrates how a single low-cost sensor (<200$) in combination with a simple yet robust and effective data transformation for CNNs yields a real-time capable HGR system. It furthermore brings along the desirable features of illumination invariance and quick sensor calibration due to the way features are extracted. Further work will address the applicability of the approach to include dynamic gestures as well as the optimization of the system for the given scenario.

REFERENCES

[1] T. Kopinski, S. Magand, A. Gepperth, and U. Handmann, "A pragmatic approach to multi-class classification," in *The International Joint Conference on Neural Networks (IJCNN 2015)*, Killarney, Ireland, 2015, pp. 1–8. [Online]. Available: http://ieeexplore.ieee.org/document/7280768/

[2] T. Kopinski, A. Gepperth, and U. Handmann, "A time-of-flight-based hand posture database for human-machine interaction," in *14th International Conference on Control, Automation, Robotics and Vision, ICARCV 2016*, Phuket, Thailand, 2016. [Online]. Available: http://ieeexplore.ieee.org/document/7838613/

[3] A. Ramamoorthy, N. Vaswani, S. Chaudhury, and S. Banerjee, "Recognition of dynamic hand gestures," *Pattern Recognition*, vol. 36, no. 9, pp. 2069–2081, 2003.

[4] T. Kopinski, A. Gepperth, and U. Handmann, "A real-time applicable dynamic hand gesture recognition framework," in *IEEE 18th International Conference on Intelligent Transportation Systems (ITSC 2015)*. Las Palmas de Gran Canaria, Canary Islands (Spain): IEEE, 2015, pp. 2358 – 2363. [Online]. Available: http://ieeexplore.ieee.org/document/7313473/

[5] A. Kurakin, Z. Zhang, and Z. Liu, "A real time system for dynamic hand gesture recognition with a depth sensor," in *Signal Processing Conference (EUSIPCO), 2012 Proceedings of the 20th European*. IEEE, 2012, pp. 1975–1979.

[6] M. Van den Bergh and L. Van Gool, "Combining rgb and tof cameras for real-time 3d hand gesture interaction," in *Applications of Computer Vision (WACV), 2011 IEEE Workshop on*. IEEE, 2011, pp. 66–72.

[7] T. Kopinski, A. Gepperth, and U. Handmann, "A simple technique for improving multi-class classification with neural networks," *European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, 2015.

[8] E. Kollorz, J. Penne, J. Hornegger, and A. Barke, "Gesture recognition with a time-of-flight camera," *International Journal of Intelligent Systems Technologies and Applications*, vol. 5, no. 3, pp. 334–343, 2008.

[9] M. Tang, "Recognizing hand gestures with Microsoft's kinect," *Web Site: http://www.stanford.edu/ class/ee368/Project_11/ Reports/Tang_Hand_Gesture_Recognition. pdf*, 2011.

[10] T. Kopinski, A. Gepperth, S. Geisler, and U. Handmann, "Neural network based data fusion for hand pose recognition with multiple ToF sensors," in *Lecture Notes in Computer Science: Artificial Neural Networks and Machine Learning ICANN 2014*, vol. 8681. Springer Verlag, Berlin, Heidelberg, Germany, 2014, pp. 233–240.

[11] T. Kopinski, A. Gepperth, and U. Handmann, "A simple technique for improving multi-class classification with neural networks," in *23th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN 2015)*, Bruges, Belgium, 2015, pp. 469–474. [Online]. Available: https://www.elen.ucl.ac.be/Proceedings/esann/esannpdf/es2015-136.pdf

[12] R. Glatt, "Deep learning architecture for gesture recognition," 2014.

[13] P. Barros, G. I. Parisi, D. Jirak, and S. Wermter, "Real-time gesture recognition using a humanoid robot with a deep neural architecture," in *Humanoid Robots (Humanoids), 2014 14th IEEE-RAS International Conference on*. IEEE, 2014, pp. 646–651.

[14] A. Tang, K. Lu, Y. Wang, J. Huang, and H. Li, "A real-time hand posture recognition system using deep neural networks," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 6, no. 2, p. 21, 2015.

[15] J. Bergstra, O. Breuleux, F. Bastien, P. Lamblin, R. Pascanu, G. Desjardins, J. Turian, D. Warde-Farley, and Y. Bengio, "Theano: a CPU and GPU math expression compiler," in *Proceedings of the Python for Scientific Computing Conference (SciPy)*, Jun. 2010, oral Presentation.

[16] F. Bastien, P. Lamblin, R. Pascanu, J. Bergstra, I. J. Goodfellow, A. Bergeron, N. Bouchard, and Y. Bengio, "Theano: new features and speed improvements," Deep Learning and Unsupervised Feature Learning NIPS 2012 Workshop, 2012.

[17] T. Kopinski, S. Geisler, L.-C. Caron, A. Gepperth, and U. Handmann, "A real-time applicable 3D gesture recognition system for Automobile HMI," in *IEEE Conference on Intelligent Transportation Systems (ITSC 2014)*, Qingdao, China, 2014, pp. 2616–2622. [Online]. Available: http://ieeexplore.ieee.org/document/6958109/

[18] T. Kopinski, J. Eberwein, S. Geisler, and U. Handmann, "Touch versus mid-air gesture interfaces in road scenarios - measuring driver performance degradation," in *IEEE 19th International Conference on Intelligent Transportation Systems (ITSC 2016)*. Rio de Janeiro, Brazil: IEEE, 2016, pp. 661–666. [Online]. Available: http://ieeexplore.ieee.org/document/7795624/