End-to-end Prediction of Driver Intention using 3D Convolutional Neural Networks

Patrick Gebert*, Alina Roitberg*, Monica Haurilet and Rainer Stiefelhagen
Karlsruhe Institute of Technology, Germany
{firstname.lastname}@kit.edu

***Abstract*— Despite extraordinary progress of Advanced Driver Assistance Systems (ADAS), an alarming number of over 1,2 million people are still fatally injured in traffic accidents every year[1]. Human error is mostly responsible for such casualties, as by the time the ADAS system has alarmed the driver, it is often too late. We present a vision-based system based on deep neural networks with 3D convolutions and residual learning for anticipating the future maneuver based on driver observation. While previous work focuses on hand-crafted features (*e.g.* head pose), our model predicts the intention directly from video in an end-to-end fashion. Our architecture consists of three components: a neural network for extraction of optical flow, a 3D residual network for maneuver classification and a Long Short-Term Memory network (LSTM) for handling temporal data of varying length. To evaluate our idea, we conduct thorough experiments on the publicly available Brain4Cars benchmark, which covers both inside and outside views for future maneuver anticipation. Our model is able to predict driver intention with an accuracy of 83,12% and 4,07s before the beginning of the maneuver, outperforming state-of-the-art approaches, while considering the inside view only.**

I. Introduction and Related Work

Most road fatalities are caused by inappropriate vehicle maneuvers due to human error [1], [2]. Advanced Driver-Assistance Systems (ADAS) identify critical events mostly *after* they have been induced by the driver and therefore following the onset of a dangerous situation. The critical time slot between recognition of a potentially dangerous action by applied ADAS and the resulting collision is often too short for a proper reaction by the vehicle systems. How can we foresee dangerous situations earlier?

Timely anticipation of driver intention offers a possible solution for allowing ADAS to prevent potential accidents at an early stage. This made vehicle maneuver prediction an established research topic in the last decades, mostly addressed by classifying hand-crafted features with a variety of approaches, such as Support Vector Machines [3], Relevance Vector Machines [4], Hidden Markov Models [5] and Recurrent Neural Networks [6]. Despite recent progress in end-to-end deep learning, the vast majority of previous work on driver maneuver anticipation has been based on manually designed feature descriptors, often employing eye gaze, head and body pose or context features, such as GPS and car speed [7], [5], [6]. One significant bottleneck of end-to-end architectures for real-life applications is their sensitivity to the amount of training data, as computer vision datasets often contain millions of training examples [8], while driver observation benchmarks, such as Brain4Cars [5] contain only hundreds.

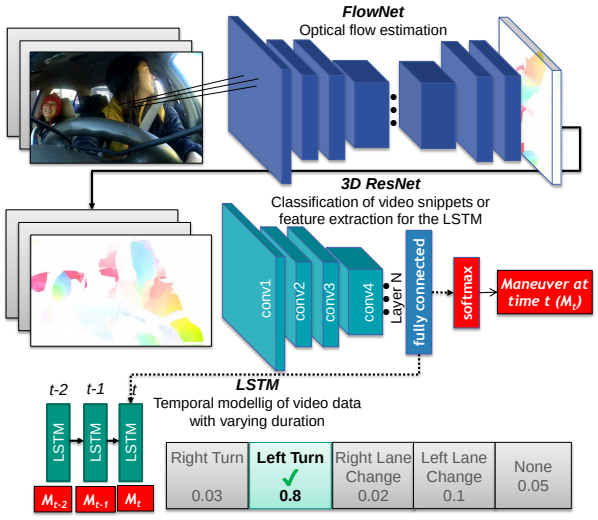
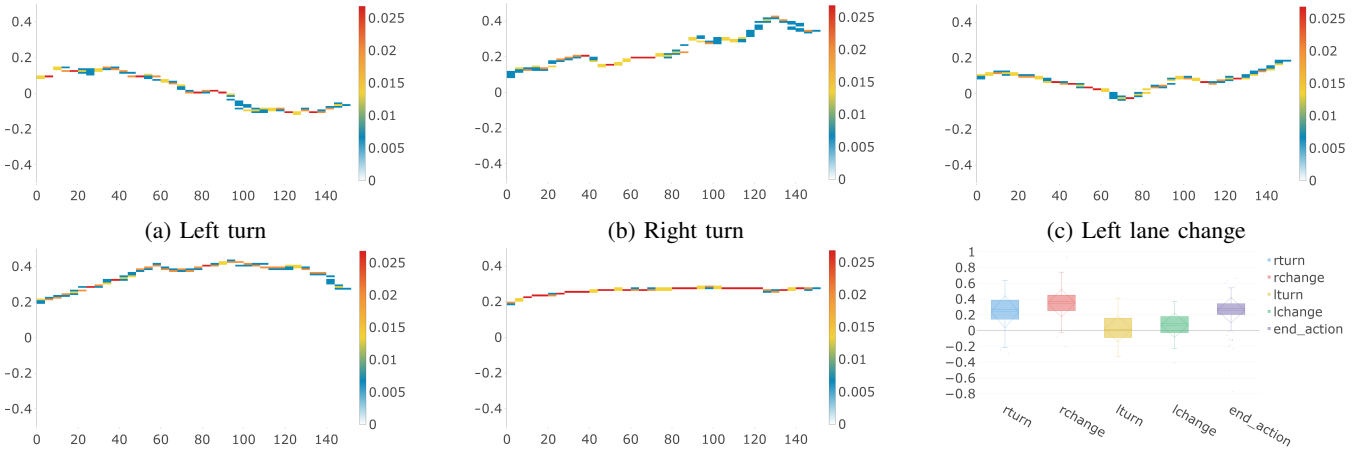


Fig. 1: Overview over the proposed neural network-based framework for anticipating the future driver maneuver.

To a large extent, anticipating driver intention is a computer vision and video analysis problem, since motion patterns such as body language and eye gaze are promising cues for predicting future actions. Recent emergence of deep learning methods has revolutionized the field, shifting the recognition paradigm from explicit definition of feature descriptors by hand (*e.g.* body pose) [9] to *end-to-end learning* of good representations directly from visual input through Convolutional Neural Networks (CNNs) [10], [11], [12]. Modern architectures for video analysis, such as action recognition, usually derive from image-based models, where the core classification is applied to video frames and extended to the temporal dimension [12], [11], [13], [14], [15]. Optical flow is sometimes used instead of or in addition to the raw RGB videos, in order to obtain a motion-based representation [12], [11]. There are different strategies for handling the temporal dimension with classifying image frames with conventional 2D CNNs and then averaging the results of all frames [11], placing a recurrent neural network, such as an LSTM, on top of the CNN [13], [16]

*Authors contributed equally to this work and are listed alphabetically.
[1]The World Health Organization (WHO)



(a) Left turn (b) Right turn (c) Left lane change

(d) Right lane change (e) No induced maneuver (f) Box plot of mean yaw angle

Fig. 2: Analysis of the drivers' motion prior to different maneuvers. **Figures 2a-2e** are a 2D-histogram visualizations of the head movement trajectory distribution 6s before the event was induced (statistics calculated over the Brain4Cars dataset using multiple drivers). The X and Y axes show the time frame and the yaw angle of the drivers' head respectively, the color stands for the frequency in the corresponding 2D-bin. **Figure 2f** summarizes the mean head pose 6s prior to each maneuver as a boxplot. The results show that head motion patterns are characteristic for each future maneuver type.

or learning spatiotemporal features through 3D convolution filters [14], [12]. Recently, Hara *et al.* [15], [17] suggested employing residual connections to build a very deep 3D CNN architecture (3D ResNet), achieving excellent results for action classification from RGB videos. We leverage the findings of Hara *et al.* [15] for driver intention prediction and propose an end-to-end architecture, which employs a 3D ResNet and extends it with an optical flow extraction network FlowNet 2.0 [18] and an LSTM for handling video data of variable duration.

Contributions and Summary

We aim to leverage the recently emerged computer vision approaches for end-to-end video analysis and present a deep-learning based framework for driver's intention prediction. Our model comprises three components: a neural network for estimating optical flow, a very deep video classification network based on 3D convolutions and residual connections and an LSTM for handling data of varying input length (overview in Figure 1). We apply our method to both, driver observation data from inside the vehicle cabin and the street view data from an outside camera by fusing both sources via a multi-stream network.

In order to deal with the limited amount of training data in naturalistic driving datasets, we pre-train our model on the Kinetics dataset [12] and transfer the learned structures to our application, further fine-tuning the network for our task. We demonstrate the effectiveness of our approach on the publicly available Brain4Cars [5] benchmark, being able to predict the maneuver with an accuracy of $83,12\%$ and $4,07s$ in advance, surpassing the current state-of-the-art approaches while using the inside view only. Our experiments show encouraging evidence that diver monitoring approaches could benefit further from modern transfer- and deep learning methods for video classification, which *learn* the representation directly from the image input, instead of modeling the input features explicitly by hand.

II. Deep Driver Maneuver Prediction with 3D Convolutional Neural Networks

In this work, we tackle the problem of vehicle maneuver prediction from video data, which aims to find the future driving event based on a given video sample. To this end, we employ a 101-layered deep neural architecture, which makes use of 3D convolutions in combinations with residual connections. Conceptually, our maneuver prediction model (see Figure 1) consists of three components: 1) an optical flow extraction network, 2) a convolution neural network based on 3D convolutions with residual connections for classification of the maneuver label (see Figure 4) and 3) an optional LSTM for handling video data of variable lengths.

Next, we provide a formal definition of the driver maneuver anticipation task (Section II-A) and describe the network components of the proposed model (Section II-C).

A. Maneuver Anticipation

The scenario that we are considering is that a person is behind the steering wheel inducing a certain driving event (*e.g.* a left turn) at time point T while being monitored with a video camera. Our goal is to correctly predict the executed maneuver *before* it took place. More formally, let $(x_0,...,x_{T-1})$ be an available video sequence with a total number of T frames. We aim to assign a maneuver label m_t to the video frames $(x_0,...,x_t)$ that precede the driver action by $T-t$, where $t<T$.

Fig. 3: Optical flow visualization of motion inside the cabin prior to a *left turn* (output of our first component).

The basic task consists of classifying the complete observation sequence from the start time of the video up to the last point before the action took place $t = T - 1$. Since the core application idea of our model is to intervene, when the human is about to induce a dangerous maneuver, it is useful to predict the maneuver *earlier* than $T - 1$ and therefore restrain the video snippet to earlier time points. The difficulty of our maneuver prediction task is dependent on the duration to the next maneuver (*i.e.* it is more challenging for smaller t). In this work, we consider both tasks: 1) maneuver prediction directly before a maneuver event ($t = T - 1$) and 2) assessments up to 6 seconds prior to the maneuver ($t < T - 1$).

B. Connection between Motion and Maneuvers

Since inducing a car maneuver requires *active* effort as the driver often performed strong movements (*e.g.* looking over the shoulder to check for bypassing vehicles), we set the focus on movement-based cues instead of still images. To validate this assumption, we first conduct a statistical analysis of motion differences for different maneuvers inside the vehicle cabin. Figures 2a-2e depict the distribution of the drivers' *yaw head angle* five seconds before the maneuver extracted from the Brain4Cars dataset, which covers ten different people[5]. The histograms illustrate that the drivers' average head motion forms a temporal pattern depending on the current maneuver.

Figure 2f summarizes the statistics of the head yaw angle prior to the driving event, showing that the head is often turned in the direction of the maneuver. The lane change maneuver is also clearly linked to the head motion but the characteristic pattern is softer than for the turns. Furthermore, constructing a simple threshold classifier based on the yaw angle of the last 6 seconds only, allows us to correctly assign future maneuver $37,6\%$ of the times, which is clearly above random chance (20%). This statistical analysis re-assures the significance of human motion as a discriminative feature, as the driver is preparing to execute a vehicle action.

C. Neural Architecture

Motion Representation. Inspired by the analysis in the previous section, we make use of a *motion* representation of the video by employing an *optical flow* extraction neural network [18] as the first component of our architecture. The optical flow obtained from consecutive frames describes a displacement vector in the X- and Y-direction for each pixel in the image. We use the state-of-the-art architecture FlowNet 2.0 [18] and transform the obtained displacement optical flow in X- and Y-direction into the RGB space. For this, we employ the publicly available PyTorch implementation of [19] pre-trained on the Sintel dataset [20]. An example output of this first step is illustrated in Figure 3, where an RGB visualization of optical flow fields prior to a *left turn* shows a person turning the head to the left and back.

Maneuver Classification with 3D ResNets. The main component of our system is a convolutional neural network (CNN) with spatiotemporal 3D kernels, as we also need to handle the temporal dimension. This neural network takes as input an optical flow video snippet of 16 frames and learns to predict the future maneuver. Our framework is based on the ResNeXt-101 architecture – a 3D convolutional residual neural network for action recognition proposed by Hara *et al.* [15]. The 101-layered 3D convolutional architecture consists of an ensemble of shallow ResNeXt blocks – a series of convolution layers with ReLu and batch normalization which are bypassed by an identity mapping connecting the previous layer to the output of the convolution layers. Both signals are then combined via summation, followed by a ReLU nonlinearity function. The advantage of *residual* networks mostly come from the residual connection, which is effective against degradation in case of very deep networks.

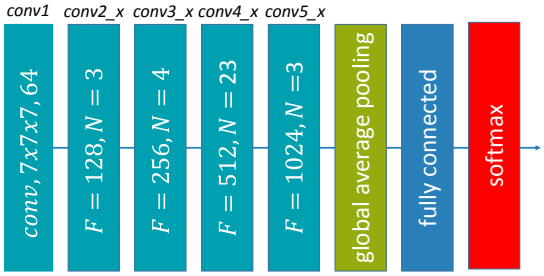


Fig. 4: The 3D ResNeXt network architecture combining 3D convolutions and residual learning. F refers to the number of filter maps and N is the number of ResNeXt building blocks used in the corresponding network stage.

Each ResNeXt block consists of three 3D convolutional layers and utilizes a *group convolution* in the middle layer. Group convolution divides the feature maps into smaller groups with a cardinality of 32. Each layer is followed by a batch normalization and a rectified linear unit (ReLU). The complete ResNeXt-101 network (see Figure 4) consist of an initial convolutional layer with $7 \times 7 \times 7$ kernels and 64 feature channels, followed by 3 ResNeXt blocks with 128 channels, 4 blocks with 256 channels, 23 blocks with

512 channels and 3 blocks with 1024 channels. The resulting 2048 channels are combined in a global average pooling layer, followed by a fully connected with a softmax activation and C hidden units, where C corresponds to the number of classes (in our case $C = 5$).

Since the datasets for driver intention prediction are too small for training end-to-end models from scratch, we use a model pre-trained on the large-scale Kinetics dataset for human activity recognition [12]. Then, we fine-tune the model on the optical flow visualization of the Brain4Cars video samples, which are always 150 frames long.

The temporal data augmentation clips the sample to 16 frames at a random position. The spatial data augmentation performs a random multi scale corner crop. It resizes the cropped samples to 112×112 pixels and flips the samples horizontally by chance. We train the network with cross-entropy loss and stochastic gradient descent from scratch with a learning rate of 0.1 which is divided by 10 after the validation loss saturates. We apply a weight decay of 0.001 and a momentum of 0.9.

LSTM for Handling Variable Video Lengths. While the proposed CNN architecture described in the previous section has excellent classification performance when predicting the maneuver from fixed-sized videos, it is unable to handle time series of varying length. In a real-world application a good model needs to predict the maneuver as soon as possible. Sometimes the cues for the future action are visible six seconds before, while in other cases, the drivers' intention might not be visible until very close to the event execution. Ideally, our model would be able to continuously handle incoming data and predict the maneuver at different time steps.

To deal with varying input sizes, we combine the 3D ResNet architecture with a Long Short-Term Memory (LSTM) network [21]. LSTM is a recurrent neural network (RNN) architecture based on a gated network cell unit. Gate units regulate the data stream through this cell unit and control the internal network state. A basic cell unit comprises an *input*, *forget* and *output gate* which control the amount of input information to be stored and thus the amount of past knowledge that can be "forgotten". Contrary to RNNs, the cell gates enforce a constant error flow back through the network graph and avoids the problem of vanishing (or exploding) gradients. LSTMs can bridge long time intervals without losing the ability to learn short time dependencies.

The mathematical formulation of an LSTM network with a gated cell unit looks as follows:

$$
\begin{aligned}
i_t &= \sigma(W_{ii} \cdot x_t + b_{ii} + W_{hi} \cdot h_{t-1} + b_{hi}) \\
f_t &= \sigma(W_{if} \cdot x_t + b_{if} + W_{hf} \cdot h_{t-1} + b_{hf}) \\
g_t &= \tanh(W_{ig} \cdot x_t + b_{ig} + W_{hg} \cdot h_{t-1} + b_{hg}) \\
o_t &= \sigma(W_{io} \cdot x_t + b_{io} + W_{ho} \cdot h_{t-1} + b_{ho}) \\
c_t &= f_t c_{t-1} + i_t g_t \\
h_t &= o_t \tanh(c_t)
\end{aligned}
$$

where i_t, f_t and o_t are the input, forget and output gate of time step t, respectively. The hidden state h_{t-1} is used together with the present input sample x_t to produce g_t at time step t. The input gate i_t regulates the addition of the input candidate information to the past cell state c_{t-1} while the forget gate f_t controls the loss of cell state information. The output gate o_t modulates the hidden state h_t which is the output of the network at time step t. W, b and σ refer to weight matrices, biases, and the sigmoid function, respectively.

The last fully connected layer of the ResNet is used as an input to the LSTM network with two layers and with 30 hidden units. Then, a fully connected layer is used with the number of neurons set to the number of maneuver classes (five for the Brain4Cars dataset), which is then normalized by a softmax layer that models the class probability distribution.

The input is split up into blocks of 25 frames which get passed to the LSTM at each time step. The LSTM network is trained jointly with the 3D ResNet using stochastic gradient descent for 150 epochs and an initial learning rate of 0.001 which gets divided by 10 after the validation loss saturates. The loss is calculated after every time step in order to ensure both late and early predictions. We use a momentum of 0.9 and a dropout rate of 0.5.

Fusing both Views via Multi-Stream Nets. In the next step, we aim to investigate and compare the usefulness of inside and the outside view. While our work primarily focuses on driver movement and is inspired by architectures used for human activity classification, the context outside of the car might offer useful cues for maneuver prediction.

First, to obtain direct comparison to the outside view, we apply the training procedure with the proposed architecture on the outside view only. Then, we leverage the multi-stream paradigm [11] to learn a joint representation for both data streams and leverage the diverse information for the prediction. The multi-stream architecture passes the inside and outside data flows through to separate networks and joins them via late fusion. Each stream is based on the single-modality architecture based on optical flow and 3D ResNet described in the previous section. To incorporate both modalities, we replace the global average pooling layers architecture with two fully connected layers with 4096 neurons fusing both input streams. Then, we apply classification fully connected layers and the final softmax normalization. We train both streams first separately on each view and, then, we fine-tune them together to learn a joint representation of the two data sources.

III. Experiments

A. Evaluation setting

We evaluate our model on the publicly available Brain4Cars benchmark [5]. Brain4Cars is a naturalistic driving dataset for driver maneuver prediction, covering both videos inside and outside the vehicle cabin 6 seconds prior to the maneuver[2]. The inside video data captures the frontal

[2]The Brain4Cars dataset is available at http://brain4cars.com. Please note, that a part of training data is missing (only 594 of the reported 700 videos are made available).

Method	Inside	Outside	*Acc* [%]	±*SE*	F_1 [%]	±*SE*
Baseline Methods						
Chance	–	–	20,0	–	20,0	–
Prior	–	–	39,0	–	–	–
Methods from [5] and [6]						
IOHMM	✓	✓	–	–	72,7	–
AIO-HMM	✓	✓	–	–	74,2	–
S-RNN	✓	✓	–	–	74,4	–
F-RNN-UL	✓	✓	–	–	78,9	–
F-RNN-EL	✓	✓	–	–	80,6	–
Our 3D-ResNet-based Architecture						
Outside-only		✓	53,2	± 0,5	43,4	± 0,9
Inside-only	✓		**83,1**	± 2,5	**81,7**	± 2,6
Two Stream	✓	✓	75,5	± 2,4	73,2	± 2,2

TABLE I: Zero time-to-maneuver results: accuracy and F_1 score computed of different models on the Brain4Cars dataset using 5-fold cross-validation (mean and standard deviation over the folds). The approaches in the second group are the best performing models of Jain *et al.* [5], [6].

view of the driver, while the outside camera faces the street scene ahead. Originally, [5] reports that the dataset compromises 700 vehicle maneuvers sampled from 10 test subjects. However, we found that a portion of training data is missing and only 594 maneuver videos are publicly available: 234 videos of driving straight, 124 videos of left lane change, 58 videos of left turn, 123 videos of right lane change and 55 videos of right lane change. Following the experimental setting of Jain *et al.* [5], we evaluate our model using 5-fold cross validation. In the final results, we report the mean and standard deviation over the five test sets, while one of the four remaining splits were used as a validation set for the hyper-parameter tuning.

B. Metrics

We evaluate our approach with two metrics: the classification accuracy, handling the maneuvers and driving straight as a separate class, and the F_1 score for detecting the maneuvers (*i.e.* the harmonic mean of the precision and recall as also used by related work [5]).

The accuracy metric is defined as follows:

$$Acc := \frac{1}{n}\sum_{i=1}^{n}\sigma(p(s_i),t_i), \text{where } \sigma(i,j) := \begin{cases} 1 & \text{if } i=j \\ 0 & \text{otherwise} \end{cases}$$

where p refers to the prediction of the classifier for the sample s_i with corresponding target label t_i and n the total number of samples in the data set.

To measure the performance of their models, Jain *et al.* [5] define precision and recall as follows:

- true prediction (tp): correct prediction of the maneuver
- false prediction (fp): prediction is different than the actual performed maneuver
- false positive prediction (fpp): a maneuver-action predicted, but the driver is driving straight
- missed prediction (mp): a driving-straight predicted, but a maneuver is performed

$$Pr = \frac{tp}{\underbrace{tp+fp+fpp}_{\text{Total \# of predictions}}}, \; Re = \frac{tp}{\underbrace{tp+fp+mp}_{\text{Total \# of maneuvers}}}$$

Using precision and recall, we calculate the F_1 score as:

$$F_1 = \frac{2\cdot Pr\cdot Re}{Pr+Re}$$

C. Zero time-to-maneuver prediction

In this section, we evaluate our model in the *zero time-to-maneuver* setting, where the complete videos up to the last frame before the starting point of the maneuver are used as input (see Table I). Additionally to the comparison with the previous work, which focused on hand-crafted features, we investigate our model's performance on the inside- and outside view as well as the combination of both via the multi-stream networks.

The model trained on the inside view achieves the best performance, surpassing the multi-stream architecture by 7,6% and the outside view by 29,9%. We link this surprisingly strong performance of the inside-only model to the knowledge transfer by pre-training our model using the Kinetics dataset [12] for human activity recognition, which we employed due to the small size of the Brain4Cars dataset. The Kinetics dataset [12] is highly human-centered and the transferred structures might be less applicable to the street images. Of course, another important reason for this high recognition accuracy remains the usefulness of observing the driver, as human motion and behavior patterns differ significantly in the preparation stages of different maneuvers, as we have shown in Section II-B. We want to highlight, that pre-training the 3D ResNet component on a large-scale dataset focused on classification of *outside* scenes might lead to better results for the street view variant of our model.

Our end-to-end model based on 3D convolutions and residual learning is able to predict driver intention with an accuracy of 83,12% and an F1 score of 81,74%, surpassing state-of-the-art. Note, that, at the same time, our model was optimized on 15% less training data than the reference approaches [5], [6] and did not use any additional context features, such as GPS coordinates or vehicle speed.

Maneuver	Pred. Frame	Pred. Time [s]	FTM	TTM [s]
Left turn	52,0 ± 6,7	2,1 ± 0,3	98,4 ± 6,7	3,9 ± 0,3
Left change	49,2 ± 4,3	2,0 ± 0,2	100,8 ± 4,3	4,0 ± 0,2
Right turn	56,0 ± 3,7	2,2 ± 0,2	94,1 ± 3,7	3,8 ± 0,2
Right change	40,8 ± 2,3	1,6 ± 0,1	109,2 ± 2,3	4,4 ± 0,1
End action	43,4 ± 3,3	1,7 ± 0,1	106,6 ± 3,3	4,3 ± 0,1
Avg. ± SE	48,3 ± 2,8	1,9 ± 0,11	101,7 ± 2,8	4,1 ± 0,1

TABLE II: Varying time-to-maneuver evaluation: prediction time, Time-To-Maneuver (TTM) and Frame-To-Maneuver (FTM) of the final LSTM network using 3D convolution and residual learning for feature extraction.

D. Varying time-to-maneuver prediction

Our next area of investigation is the *varying time-to-maneuver* setting, where we aim to predict driver event up to 5s earlier, taking different time steps into account. We employ an LSTM network on top of our original model to handle data input of varying durations (see Section II-C).

We adopt the *time-to-maneuver* evaluation procedure from [5] and estimate the earliest time step when a test sample is predicted correctly for the first time. On average, the maneuvers were predicted correctly 4,1 seconds before the event has been induced (reported in Table II).

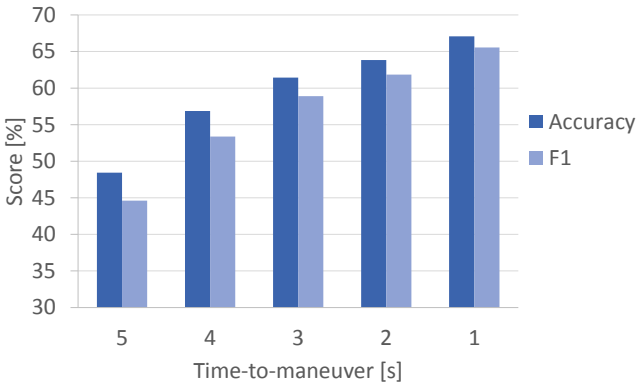


Fig. 5: The accuracy and the F_1 score of the model on the test set depending on the time-to-maneuver. The recognition rate is higher, the closer we are to the event starting point.

Additionally, to the *time-to-maneuver* analysis, we provide the accuracy and F1 score depending on the period of time before the beginning of the maneuver in Figure 5. Of course, prediction at an earlier stage is a harder task and the accuracy drops compared to the zero-time to maneuver case. Still, over 60% of events are correctly predicted 3 seconds before their occurrence, which is significantly higher than the random baseline (20% for five maneuver types).

IV. Conclusion

This work investigates the problem of anticipating future vehicle maneuvers from video data. We combine a neural network for optical flow extraction with an architecture based on 3D convolutions with residual connections consisting of 101-layers. As the amount of training data is very small, we make use of the Kinetics human action dataset to pre-train our model and further fine-tune it on the target Brain4Cars. With an overall accuracy of $83,12\%$ and an F_1 score of $81,74\%$, our model outperforms previous state-of-the-art approaches. Combined with an LSTM module, the network is able to handle input sequences of variable temporal duration and can anticipate vehicle maneuvers $4,07s$ in advance. Finally, our inside and outside view evaluation revealed that the indoor data provides better cues for predicting maneuvers at an early stage than the street view data.

Acknowledgements The research leading to this results has been partially funded by the German Federal Ministry of Education and Research (BMBF) within the PAKoS project.

References

[1] S. Reynolds, M. Tranter, P. Baden, D. Mais, A. Dhani, E. Wolch, and A. Bhagat, "Reported Road Casualties Great Britain: 2016 Annual Report," UK Department for Transport, Tech. Rep. September, 2017.

[2] S. Singh, "Critical reasons for crashes investigated in the National Motor Vehicle Crash Causation Survey," *National Highway Traffic Safety Administration*, no. February, pp. 1–2, 2015.

[3] P. Kumar, M. Perrollaz, C. Laugier, P. Kumar, M. Perrollaz, P. Kumar, and M. Perrollaz, "Learning-based approach for online lane change intention prediction To cite this version : Learning-Based Approach for Online Lane Change Intention Prediction," no. Iv, pp. 1–6, 2013.

[4] A. Doshi and M. Trivedi, "A comparative exploration of eye gaze and head motion cues for lane change intent prediction," *IEEE Intelligent Vehicles Symposium, Proceedings*, pp. 49–54, 2008.

[5] A. Jain, H. S. Koppula, B. Raghavan, S. Soh, and A. Saxena, "Car that knows before you do: Anticipating maneuvers via learning temporal driving models," *Proceedings of the IEEE International Conference on Computer Vision*, vol. 2015 Inter, pp. 3182–3190, 2015.

[6] A. Jain, A. Singh, H. S. Koppula, S. Soh, and A. Saxena, "Recurrent Neural Networks for driver activity anticipation via sensory-fusion architecture," *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 2016-June, no. Figure 2, pp. 3118–3125, 2016.

[7] E. Ohn-Bar and M. M. Trivedi, "Looking at humans in the age of self-driving and highly automated vehicles," *IEEE Transactions on Intelligent Vehicles*, vol. 1, no. 1, pp. 90–104, 2016.

[8] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. Ieee, 2009, pp. 248–255.

[9] A. Roitberg, N. Somani, A. Perzylo, M. Rickert, and A. Knoll, "Multimodal human activity recognition for industrial manufacturing processes in robotic workcells," in *Proceedings of the 2015 ACM on International Conference on Multimodal Interaction*. ACM, 2015.

[10] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[11] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," in *Advances in neural information processing systems*, 2014, pp. 568–576.

[12] J. Carreira and A. Zisserman, "Quo vadis, action recognition? a new model and the kinetics dataset," in *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*. IEEE, 2017.

[13] J. Y.-H. Ng, M. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, and G. Toderici, "Beyond short snippets: Deep networks for video classification," in *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on*. IEEE, 2015, pp. 4694–4702.

[14] S. Ji, W. Xu, M. Yang, and K. Yu, "3d convolutional neural networks for human action recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 1, pp. 221–231, 2013.

[15] K. Hara, H. Kataoka, and Y. Satoh, "Can Spatiotemporal 3D CNNs Retrace the History of 2D CNNs and ImageNet?" 2017.

[16] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell, "Long-term recurrent convolutional networks for visual recognition and description," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015.

[17] K. Hara, H. Kataoka, and Y. Satoh, "Learning spatio-temporal features with 3d residual networks for action recognition," in *Proceedings of the ICCV Workshop on Action, Gesture, and Emotion Recognition*, vol. 2, no. 3, 2017, p. 4.

[18] A. Dosovitskiy, P. Fischery, E. Ilg, P. Hausser, C. Hazirbas, V. Golkov, P. V. D. Smagt, D. Cremers, and T. Brox, "FlowNet: Learning optical flow with convolutional networks," *Proceedings of the IEEE International Conference on Computer Vision*, vol. 2015 Inter, pp. 2758–2766, 2015.

[19] F. Reda, R. Pottorff, J. Barker, and B. Catanzaro, "flownet2-pytorch: Pytorch implementation of FlowNet 2.0: Evolution of Optical Flow Estimation with Deep Networks," 2017.

[20] N. Mayer, E. Ilg, P. Hausser, P. Fischer, D. Cremers, A. Dosovitskiy, and T. Brox, "A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 4040–4048.

[21] S. Hochreiter and J. Urgen Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.