Body Pose and Context Information for Driver Secondary Task Detection

Manuel Martin[1], Johannes Popp[2], Mathias Anneken[2], Michael Voit[1] and Rainer Stiefelhagen[2]

***Abstract*— Distraction of the driver by secondary tasks is already dangerous while driving manually but especially in handover situations in an automated mode this can lead to critical situations. Currently, these tasks are not taken into account in most modern cars. We present a system that detects typical distracting secondary tasks in an efficient modular way. We first determine the body pose of the driver and afterwards use recurrent neuronal networks to estimate actions based on sequences of the captured body poses. Our system uses knowledge about the surroundings of the driver that is unique to the car environment. Our evaluation shows that this approach achieves better results than other state of the art systems for action recognition on our dataset.**

I. INTRODUCTION

Advances in vehicle automation have made the task of driving a vehicle less demanding. In general this leads to less accidents. However if drivers are less occupied with the main task they tend to occupy themselves with additional secondary tasks [1]. This can include road legal tasks like using the built in infotainment system but may also include other tasks that are not allowed like typing on a phone. Depending on the automation level and the engagement of the driver in the secondary task this can lead to safety critical events. If the car could detect such secondary tasks it could however react accordingly by for example adapting the distance to the vehicle in front in the case of an adaptive cruise control or by warning earlier in a lane departure warning system. In severe cases the car could also warn the driver to stop the secondary task.

The challenges with this kind of behavior increase in automated cars of SAE Level 3 and 4 because the driver is temporarily completely relieved of the driving task [2]. However the driver still serves as a fallback and has to take over the driving task in a timely manner on system boundaries. This takeover request can occur due to an uncertain situation the automation cannot handle (SAE level 3) or at the end of the operation domain (SAE level 4). The time drivers need to take over after a request is subject to current research. There are different factors that influence the take over time. Next to the complexity [3] and the criticality [4] of the traffic situation the state of the driver is an important parameter of the takeover process. However the state of the driver can vary greatly because unlike to SAE level 2 the driver is no longer obliged to supervise the automation in higher levels. He therefore can pursue more complex secondary tasks for a longer time which can affect the take over time greatly. This should therefore be detected and factored in by the automation when planning a take over [5].

[1]Fraunhofer IOSB, Karlsruhe, Germany
manuel.martin@iosb.fraunhofer.de
[2]Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany
rainer.stiefelhagen@kit.de

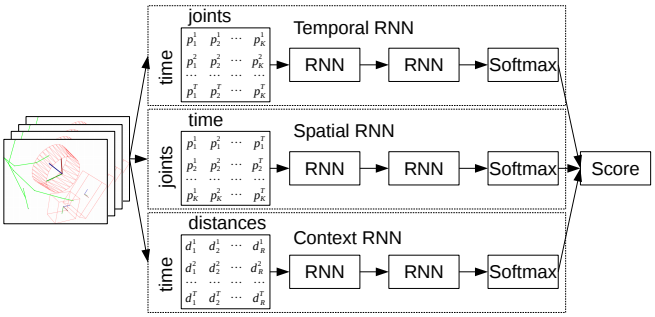


Fig. 1: Overview of the three-stream RNN architecture for skeleton and context based action recognition. Softmax denotes a fully connected layer with a softmax activation function.

We present a system that can detect secondary tasks of drivers. It is based on a state of the art approach on action recognition on sequences of 3D body poses. We extend this approach by also including information about the well known car interior and the interactions of the driver with this interior while performing secondary tasks. Figure 1 shows the overview of our end-to-end trainable architecture. To evaluate this approach we collected a dataset of test subjects performing 6 different secondary actions while driving on a test track. Some of these actions are very similar. We therefore evaluate our approach in two different ways. First, we train and test our model for action recognition where the task only consists of classifying short sequences of single tasks to evaluate how well our system can discern the classes. In a second evaluation we train our model on the more difficult detection task on long sequences containing all tasks and transitions. This demonstrates the performance of our system for real world application.

II. RELATED WORK

The goal of many optical driver observation systems is to detect what the driver is doing. However this can be achieved in different ways and with different granularity. Many approaches focus on features that can be estimated from the drivers head. Head and eye movement patterns can for example be used to distinguish between an alert driver and a distracted driver [6], [7], [8]. For a more detailed analysis of the driver's visual focus of attention an eyetracker can be used to intersect the gaze ray with the scene to find

out where the driver is looking at [9]. Based on such a system it can be determined if the driver's eyes are on the road or if he looks at something in the interior of the car [10].

However for many tasks the hands are used even without looking at a target. Eyetracking alone does not work in these cases. There are different approaches that detect the hands in color images to determine what the driver is doing [11]. This allows, for example, to detect if both hands are on the steering wheel or if one hand is at the gear stick or the infotainment controls. Often this approach is limited to a small number of regions [12], [13]. Martin et. al. show how the upper body pose of the driver can be estimated on depth camera images and how it can be used to detect the interaction with a wide range of areas in a car's interior [14].

Secondary tasks especially in automated vehicles can be much more complex than interactions with built in controls. Even today, phone use in the car is very common even if it is illegal. There are three datasets used for more complex secondary task detection [15]. All consist of still images of representative postures of the driver for different tasks like using a cell phone, eating or drinking. Many of the older approaches evaluated on these datasets use handcrafted features together with classifiers like random forests [16]. These methods already achieve very good results. Newer methods use convolutional neuronal networks solving these datasets almost perfectly [17], [18]. However these approaches are simpler compared to state of the art approaches on general action recognition while still achieving better results. This indicates that either action recognition in cars is easier than general action recognition or that the used datasets are to limited.

Datasets for general action recognition mostly consist of video sequences. Compared to the in car datasets frame based classification is not enough to achieve good results there. It is necessary to exploit temporal information to resolve ambiguities. Popular approaches use for example 3D convolutional networks applied to image stacks [19] or recurrent networks applied on image sequences [20]. Another popular research area in action recognition is the classification of sequences of 3D-body-poses. First the 3D-body-pose is extracted from image data or acquired with a motion capture system and afterwards these body pose sequences are classified for example with recurrent neuronal networks [21]. Current approaches also try to integrate the relationships between single joints directly into the recurrent networks [22], [23].

Compared to the datasets and approaches used in the state of the art for secondary task estimation in cars, we collected a larger dataset consisting of continuous video sequences with a wide variety of body poses that occur naturally when driving for a longer time in addition to different secondary tasks. This data is more challenging because it also contains transitions and other ambiguities where only the history of the performed movement allows to classify the data correctly. Additionally the recorded data contains both infrared and depth images allowing systems trained on this dataset to work at night and by day which make it more applicable in the real world compared to system that are based on color images. We use the depth data to estimate the 3D body pose of the driver and we extend a current state of the art model for general action recognition [22] to use additional knowledge about the car's interior.

III. RNNs for action detection and recognition

Most actions take a certain period of time and can be modelled by a time series of steps. Recurrent neuronal networks are especially suited for this task. A basic recurrent neuronal network layer can be defined in the following way:

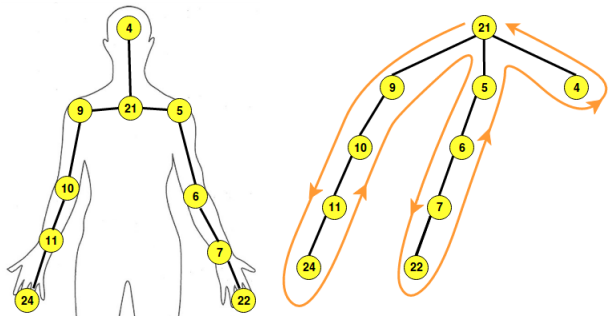
$$h_t = f(Ux_t + Vh_{t-1} + b) \tag{1}$$

Where x_t is the input vector at time t, U and V are weight matrices, b is a bias vector, f is a nonlinear function and h_t is the output of the recurrent network layer at time t. The output of the network in the current timestep therefore depends both on the current input and on the output of the network in the last timestep.

Often these networks are used with a fixed time horizon n of at most a few hundred steps. In the first timestep of an evaluation $t = 0$ the previous output h_{-1} is unknown and is often initialized with zero. When training such a network the gradient is backpropagated through all n timesteps. This approach is called a "stateless recurrent neuronal network"

Allthough networks with a fixed time horizon are often used for action recognition this has different drawbacks. If the actions that should be recognized vary in length it is not clear how long the time horizon should be. If the time horizon is too short long actions cannot be processed from start to end by the network. If the time horizon is too long padding is necessary for the remaining steps of the network. Most large datasets for action recogntion at the time of writing consist of short clips that only contain a single action. If a clip is longer than the time horizon it is often truncated at the end and if the clip is shorter than the time horizon it is pre zero padded for the first steps. To evaluate such a network on longer continuous sequences a sliding window approach has to be used. This costs a lot of performance as $n-1$ previous time steps have to be reevaluated for every new datum of the sequence.

Recurrent neuronal networks can also be used with a variable number of steps. Instead of stopping the evaluation after n steps the evaluation continues as long as there is new data. For a long sequence the network is only advanced one step instead of resetting the network and reprocessing $n-1$ steps as would be necessary with the sliding window approach and a stateless model. This approach can be called "stateful recurrent neuronal network". If such a network is trained on long sequences of actions it can figure out the time horizon for each action by itself and it can also learn to handle transitions between different actions. While doing backpropagation it is however no longer feasible to do backpropagation from the current position in the sequence to the beginning. Instead the gradient is only backpropagated for a fixed number of steps. These recurrent neuronal networks are better suited for continuous streams of input data because



(a) The joints used in the recognition system (b) The order of joints processed by the spatial stream

Fig. 2: Joint representations used in the network.

they just need to advance a single step for each new frame without using a sliding window.

Instead of the simple recurrent model depicted here we use long short term memory cells (LSTM) because they do not suffer from gradient loss while training on longer sequences and because of their superior ability to retain information over a longer time [24]. LSTM layers can nevertheless be used in both stateful and stateless networks.

IV. Three-stream RNN

Sequences of skeletons are a good representation of the movement of the human body and can be used for efficient action recognition systems. Wang et al. [22] presented an approach that evaluates sequences of skeletons in two different streams that are combined in the end. The first stream is a temporal stream. It describes the development of movement of the whole skeleton over time. The second stream is called a spatial stream. It describes the spatial relationship between different body parts in a fixed time frame. We extend this framework with an additional third stream that makes use of the knowledge about the cars interior. We call this stream the context stream. It describes the relationship between body movements and parts of the interior of the car. The idea behind this extension is, that in a car, unlike many other environments, a lot of the structure is fixed and already known. This information should therefore be helpful for action recognition because interaction with the surrounding environment is part of many actions. For example, if the driver has both hands on the steering wheel he cannot at the same time use his smartphone or drink out of a bottle. On the other hand if the driver wants to drink out of a bottle he first has to pick it up from some storage area. The three streams, temporal, spatial, and context, are trained separately and merged in a late fusion approach by using a weighted average of the results of each stream. An overview of the network is shown in Figure 1.

In the following we first give a brief introduction of our previous work on body pose estimation and context modeling of the cars interior that we use as input for our secondary task estimation system. Afterwards, we go into detail how each of the three streams of our recurrent model works.

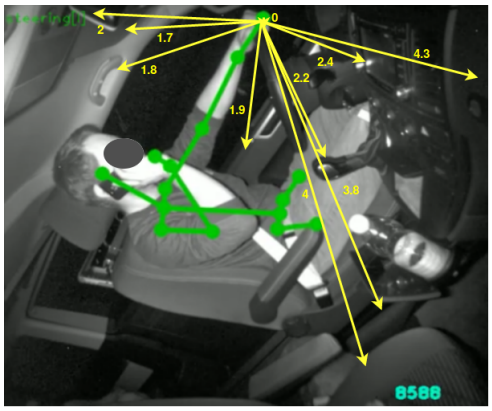


Fig. 3: Visualization of the context data represented as distances of the hands to different regions in the interior. The left hand grabs the steering wheel. The distance to the wheel is therefore zero. (best viewed in color)

A. Body pose estimation and context features in cars

Our action recognition system does rely on the 3D body pose of the driver and a model of the surrounding car interior. To generate this high level representation we use our previous work on body pose estimation [14]. It uses a depth camera and random decision forests to estimate the 3D upper body pose of the driver. In addition we also showed how a simple model of the interior consisting of shape primitives like cylinders and bounding boxes can be used to reliably determine interaction with these areas. The steering wheel for example can be represented by a cylinder while the gearstick and the infotainment area are surrounded by bounding boxes. We use this simple representation with the estimated body pose to determine the distance of the wrist joints to the surface of each primitive. In our previous work we thresholded these distances to detect interactions of the hands with these areas. For our action recognition system we do not threshold the distances but use the distances of the right and left wrist as an input vector for our recurrent model. The input to the action recognition system therefore consists of the 3D joint positions of the skeleton depicted in Figure 2a and the distances of the wrist joints to all primitives defined in the model of the car interior.

B. Temporal stream

The temporal stream models the temporal dynamics within sequences of skeletons. Similar to previous approaches [21], [22] we concatenate the 3D-joints of the skeleton in each time step to one input vector and use it as input to a RNN with two stacked LSTM layers. We also tried a hierarchical input encoding as proposed by [21]. In this case the joints of each limb and the torso are first used as input to separate LSTM layers which are then concatenated as the input for a second LSTM layer. The idea behind this hierarchy is, that different actions are focused on different body parts. So by encoding the information in a hierarchical way the LSTMs for each limb can better learn relevant actions. However we did get a performance decrease compared to stacking all

joints in a single input vector. The reason for that is most likely that we only use the upper body joints and therefore do not model any activity performed by the feet as is often the case in other more general action recognition datasets. In addition our dataset is not balanced for executions of actions with the right and left hand which makes learning of the first layer more unbalanced.

C. Spatial stream

A skeleton is a system consisting mostly of rigid limbs and joints that move in a fixed relationship. Depending on the performed action this system performs different complex patterns. To model this we use the spatial skeleton stream. Similar to [22] we perform a depth first search on the kinematic chain of the human body to generate a sequence of body joints as input for a recurrent neuronal network. The graph of body joints and the resulting order of joints in the input sequence is show in Figure 2b. The nodes in the interior of the tree are visited multiple times to create a continuous chain without broken links in the relationship between joints. Unlike the temporal and context streams the sequence length of the spatial stream is the number of joints in the skeleton and is therefore always fixed. We therefore always use a stateless RNN for this stream. The input of the network at each step consist of a vector of positions from a single joint. As a single joint only has three coordinates we use a temporal window of T steps around the current time step and concatenate the coordinates of the joint in this window as the input for the spatial stream.

D. Context stream

The context stream relies on data of the surroundings and its interaction with the movement of a person. Approaches that use whole images as input can, depending on the dataset, implicitly model some relationships of actions with the surroundings because the information is visible in the image. However to our knowledge no public datasets for skeleton based action recognition does contain such information and therefore no current approach tries to model these interactions. As described in section IV-A we have a model of parts of the interior represented as bounding volumes and we determine the distance of the wrists to each of these volumes. We stack the distances collected in one frame and use it as the input vector of another RNN with two stacked LSTM layers. The dimension of the input vector is $2R$ where R is the number of shape primitives in the model of the interior. Figure 3 shows the information extracted from one frame for the right hand.

V. Evaluation

Before we evaluate our extended model on our own dataset we verify that the two-stream-model by Wang et. al. [22] that we had to reimplement for our system works similar to the reference system. We therefore train our reimplemented system on the NTU-RGB+D dataset [25] and compare the results with the reported original performance. Afterwards we present both our own dataset and the implementation details for our models for action recognition and action detection trained on this dataset. We then discuss the results of both models.

TABLE I: The accuracy of our reimplementation of the Two-Stream RNN on the NTU-Dataset compared to the reference result from the original paper.

Model	Reimplementation	Reference
temporal	66.76	66.1
spatial	49.60	55.2
Two-Stream RNN	68.17	68.6

A. Baseline verification on the NTU-RGB+D Dataset

The approach of Wang et. al. is not open source so we had to reimplement it ourselves. To show that our implementation produces comparable results we trained and evaluated it on the NTU-RGB+D dataset with the parameters reported in the original paper. In Table I we compare our results with the results reported in the paper. The results of the temporal model are very similar. The performance of the spatial model however are worse than the original results. We could not find a reason for the difference. However there are a few design parameters for the spatial stream that are not reported in their paper and we might have implemented it slightly different. Nevertheless the result of the combined Two-Stream RNN are again comparable. We therefore conclude that our reimplementation performs comparably and we use it as our state of the art baseline on our in car dataset. We cannot test our extensions to the model on the NTU-RGB+D dataset because it does not provide the necessary context information.

B. Dataset Description

Our goal was to record a dataset that resembles a real driving scenario as close as possible with a sensor that produces data that is suitable in most lighting conditions. We therefore collected long data streams that contain multiple different actions and transitions. The data was recorded with a Microsoft Kinect for XBox One in a Volkswagen T5 bus at 30Hz. The camera was located at the a-pillar of the co-driver side looking at the driver. The Volkswagen T5 is not capable of driving in an automated mode. The drivers therefore always had to fulfill both the driving task while doing the secondary tasks. The data collection took place on a closed off airfield. An examiner sat in one of the backseats and told the test participants which secondary action to do next. The participant then had to do the action as soon as safely possible. The requested secondary actions are “drinking from a bottle”, “eating”, “using a phone for texting”, “making a call” and “reading a book”. The action “drinking from a bottle” was later split into “opening/closing a bottle” and “drinking”. The experiment took place with 28 test subjects. The resulting dataset is highly biased. 67% of the data represents just the main task of driving the car. In addition the average length of the secondary tasks varies greatly. The shortest secondary task is “opening/closing a bottle” with

80 frames on average per execution. The longest task is reading with 700 frames on average per execution. Overall the dataset contains 12 hours of annotated data. Because of this unbalanced dataset we report precision, recall and f1-score in addition to the common mean average precision used for other datasets. As a baseline we also provide the results for a naive classifier that always estimates the task driving.

C. Implementation Details

We normalized the skeleton data by subtracting the neck joint from all joints of the input sequences to make the skeleton representation location invariant. To determine the distances to the shape primitives of the interior model we use the body pose without normalization. The interior model was comprised of the 8 regions already defined and evaluated in our previous work. Namely the steering wheel, gear stick, infotainment area, hand brake, sun visor, codriver seat, glovebox and inner mirror. To determine the hyperparameters of the network we used a random search strategy. Both LSTM layers of the temporal and spatial stream have a size of 256 cells while the LSTM layers of the context stream have a size of 128 cells. The temporal window of the spatial stream was optimal with a size of 51 frames. The spatial stream uses bidirectional stateless LSTM layers for both action recognition and action detection. The temporal and context streams use bidirectional stateless LSTMs for the action recognition task with a history of 300 frames and unidirectional stateful LSTM layers for the action detection task. All networks are trained using cross entropy loss and 50% dropout. Because the dataset is unbalanced the loss of samples with a label other than driving was weighted twice as high as a sample labeled as driving. To train the system 17 streams where used for training, 2 for validation and 9 for testing.

D. Action recognition

For the action recognition task the network does not need to figure out when an action starts or ends. Instead the network is presented with a sequence of frames that consists only of a single class and it should determine which of the actions the sequence represents. To setup this task we segment our dataset on the action boundaries. Sequences that are still too long are split in smaller sequences using a sliding window approach with a stride half as long as the input length of the network. Sequences that are shorter than the input length of the network are pre zero padded. State of the art approaches for action detection use mostly stateless recurrent networks. We therefore also trained our system with a stateless recurrent network for all three streams.

Table II shows the results. Similar to the results on the NTU RGB+D dataset the temporal stream on its own works best while the spatial stream does not work well on its own. The new context stream on the other hand performs almost as well as the temporal stream. Of the possible stream combinations the state of the art two-stream RNN combining temporal and spatial streams works better that each stream on its own. However the combination of temporal and context stream already outperforms the state of the art while the combination of all three streams achieves the best results. The good performance of the context stream alone already shows that the position of the hands in the context of the interior already contains a lot of information about the performed actions.

TABLE II: The results of the stateless model for action recognition.

Model	Acc.	Precision	Recall	F-Score
naive	67.00	0	1	0
temporal	82.32	76.38	72.52	73.84
spatial	69.55	45.81	43.11	43.60
context	76.27	72.30	71.16	70.77
temporal + spatial[22]	85.63	80.92	73.36	76.39
temporal + context	85.94	81.12	**75.10**	77.22
Three Stream	**87.68**	**83.61**	74.69	**78.20**

TABLE III: The results of the stateful model for action detection

Model	Acc.	Precision	Recall	F-Score
naive	67.00	0	1	0
temporal	77.74	62.51	64.07	62.85
spatial	69.01	45.21	42.90	43.24
context	75.66	55.66	60.62	56.02
temporal + spatial[22]	81.60	68.92	64.70	66.40
temporal + context	82.21	67.76	**68.42**	**67.86**
Three Stream	**82.34**	**69.07**	66.94	67.80

Figure 4 shows the confusion matrix of the three stream model. It shows that most of the time the secondary actions are confused with the much more probable driving task. In addition the classes “phone use” and “reading” show the most consistent confusion. These are the two classes that take the longest on average and are often longer than the time frame of 300 frame of the network. This indicates that the RNN cannot detect these classes reliably if it cannot see the start of the action. This is especially true for the class reading because most of the people picked up the book and positioned it on the steering wheel. The posture after picking up the book is therefore almost identical to driving with both hands on the steering wheel.

E. Action detection

In a real world scenario the boundaries between actions are not known. There is a continuous stream of data and the system must both detect which action is taking place and also when the action is taking place. This task is more suitable for a stateful recurrent neuronal network because it can be evaluated continuously without resetting it after each evaluation.

Table III shows the results of the stateful model for action detection. Overall the behavior of each model and model fusion is similar. The temporal stream works best, followed by the context stream and the spatial stream is worst. Both models combined with the context stream are still better than the state of the art two stream model however the difference is smaller and the spatial stream does make less of

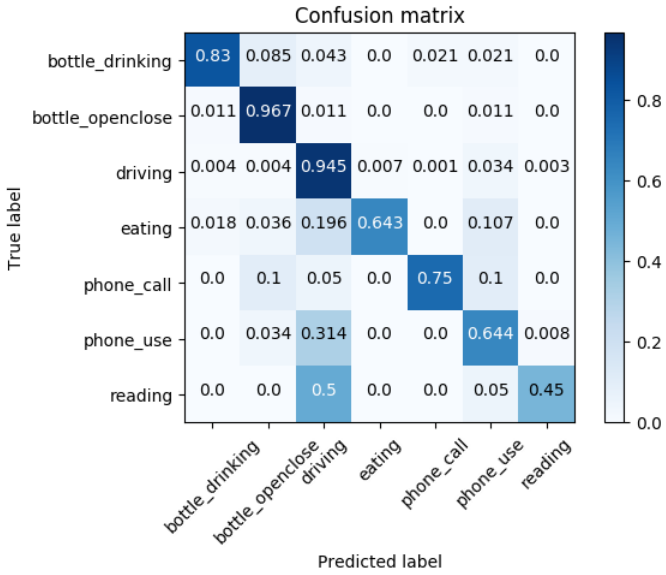


Fig. 4: Confusion matrix of the three stream stateless model for action recognition.

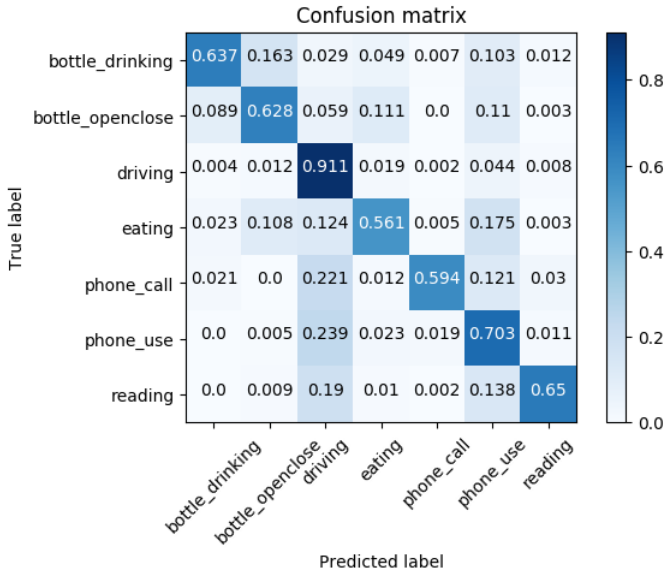


Fig. 5: Confusion matrix of the three stream stateful model for action detection.

a difference. One explanation could be that both the temporal and context streams can make use of a longer history because of their stateful recurrent networks while the spatial network is unchanged compared to the simpler recognition task and cannot benefit from a longer history.

Figure 5 shows the confusion matrix of the stateful three stream model. Compared to the confusion matrix of the model trained for action recognition alone the performance is worse for the tasks “bottle drinking”, “bottle open/close” and “phone call” while it is better for the tasks “phone use” and “reading”. The tasks the stateful model performs worse are all short on average while the tasks it performs better are longer tasks. The stateful model therefore seems to better model longer context while it has difficulty to detect shorter tasks in the stream.

The overall performance of the stateful model for detection is worse than the performance of the stateless model for action recognition. This is to be expected because it is a more difficult task. However because of the evaluation method the comparison is not entirely fair because for the action detection task the performance metrics were computed for each frame of the sequence while in the recognition task they were only determined once for the end of each short sequence. The evaluation of the detection task therefore also contains all transitions between tasks. In these cases the model often returned the class driving because it is the most frequent class in the dataset. This is not entirely wrong and it is also reflected in the greater confusion between each secondary task and the driving task in the confusion matrix. On average the stateful model needs 26 frames before it detects the correct class after a transition which is quite a fast reaction time. If these frames are excluded from the evaluation the performance metrics increase by overall 2% for each model.

Finally we also performed a short comparison of the overall runtime of the different models. The stateless three stream model for action recognition achieves a throughput of just 2 frames per second. The biggest contribution to the runtime are the temporal and context streams with a history of 300 frames which have to be computed from the start for each sequence. The stateful three stream model on the other hand already achieves 70 frames per second. Here the spatial stream contributes most to the runtime because it is still a stateless model with a fixed number of steps. The stateful model that combines the temporal and the context stream achieves 290 frames per second. It is not limited by the spatial stream and achieves only slight worse results. This shows the runtime advantage of stateful models for the action detection task.

VI. CONCLUSIONS

We proposed a method for secondary task detection based on the 3D upper body pose of the driver and the context of his surroundings. To achieve this we extended the state of the art two-stream-model for general action recognition, that combines the temporal and spatial structure of the body pose, with an additional context stream that encodes the association of the body pose and the car interior for different tasks. We evaluate the resulting system on our own challenging dataset which consists of 12 hours of video data separated into 28 long sequences of people driving while doing 6 secondary tasks. We show that both the combined temporal and context stream and the combination of all three streams outperform the original two stream model. We achieve these results while achieving 290 frames per second with a minor quality loss which makes this approach a lightweight addition to a body pose estimation system that could also be used for other purposes in a car.

REFERENCES

[1] F. Naujoks, C. Purucker, and A. Neukum, "Secondary task engagement and vehicle automation comparing the effects of different automation levels in an on-road experiment," *Transportation Research Part F: Traffic Psychology and Behaviour*, vol. 38, pp. 67 – 82, 2016.

[2] *Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles SAE J 3016*, Society of Automotive Engineers (SAE) Std., 2016.

[3] D. Damböck, M. Farid, L. Tönert, and K. Bengler, "Übernahmezeiten beim hochautomatisierten fahren," *Tagung Fahrerassistenz. München*, vol. 15, p. 16, 2012.

[4] C. Gold, D. Damböck, L. Lorenz, and K. Bengler, "Take over! How long does it take to get the driver back into the loop?" in *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, vol. 57, no. 1, 2013, pp. 1938–1942.

[5] J. Ludwig, M. Martin, M. Horne, M. Flad, M. Voit, R. Stiefelhagen, and S. Hohmann, "Driver observation and shared vehicle control: supporting the driver on the way back into the control loop," *at - Automatisierungstechnik*, vol. 66(2), pp. 146 – 159, 2018.

[6] M. Wollmer, C. Blaschke, T. Schindl, B. Schuller, B. Farber, S. Mayer, and B. Trefflich, "Online driver distraction detection using long short-term memory," *Transactions on Intelligent Transportation Systems*, vol. 12, no. 2, pp. 574–582, 2011.

[7] R. O. Mbouna, S. G. Kong, and M.-G. Chun, "Visual analysis of eye state and head pose for driver alertness monitoring," *Transactions on intelligent transportation systems*, vol. 14, no. 3, pp. 1462–1469, 2013.

[8] T. Liu, Y. Yang, G.-B. Huang, Y. K. Yeo, and Z. Lin, "Driver distraction detection using semi-supervised machine learning," *Transactions on intelligent transportation systems*, vol. 17, no. 4, pp. 1108–1120, 2016.

[9] P. Jiménez, L. M. Bergasa, J. Nuevo, N. Hernández, and I. G. Daza, "Gaze fixation system for the evaluation of driver distractions induced by ivis," *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 3, pp. 1167–1178, 2012.

[10] F. Vicente, Z. Huang, X. Xiong, F. De la Torre, W. Zhang, and D. Levi, "Driver gaze tracking and eyes off the road detection system," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 4, pp. 2014–2027, 2015.

[11] T. H. N. Le, Y. Zheng, C. Zhu, K. Luu, and M. Savvides, "Multiple scale faster-rcnn approach to drivers cell-phone usage and hands on steering wheel detection," in *Computer Vision and Pattern Recognition Workshops (CVPRW), 2016 IEEE Conference on*. IEEE, 2016, pp. 46–53.

[12] S. Y. Cheng and M. M. Trivedi, "Vision-based infotainment user determination by hand recognition for driver assistance," *IEEE transactions on intelligent transportation systems*, vol. 11, no. 3, pp. 759–764, 2010.

[13] E. Ohn-Bar, S. Martin, A. Tawari, and M. M. Trivedi, "Head, eye, and hand patterns for driver activity recognition," in *Proceedings of the International Conference on Pattern Recognition*, 2014, pp. 660–665.

[14] M. Martin, S. Stuehmer, M. Voit, and R. Stiefelhagen, "Real time driver body pose estimation for novel assistance systems," in *International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2017, pp. 1738–1744.

[15] Y. Abouelnaga, H. M. Eraqi, and M. N. Moustafa, "Real-time distracted driver posture classification," *arXiv preprint arXiv:1706.09498*, 2017.

[16] C. Yan, F. Coenen, and B. Zhang, "Driving posture recognition by joint application of motion history image and pyramid histogram of oriented gradients," *International journal of vehicular technology*, vol. 2014, 2014.

[17] ——, "Driving posture recognition by convolutional neural networks," *IET Computer Vision*, vol. 10, no. 2, pp. 103–114, 2016.

[18] S. Masood, A. Rai, A. Aggarwal, M. Doja, and M. Ahmad, "Detecting distraction of drivers using convolutional neural network," *Pattern Recognition Letters*, 2018.

[19] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-scale video classification with convolutional neural networks," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2014, pp. 1725–1732.

[20] F. J. Ordóñez and D. Roggen, "Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition," *Sensors*, vol. 16, no. 1, p. 115, 2016.

[21] Y. Du, W. Wang, and L. Wang, "Hierarchical recurrent neural network for skeleton based action recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1110–1118.

[22] H. Wang and L. Wang, "Modeling temporal dynamics and spatial configurations of actions using two-stream recurrent neural networks," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[23] J. Liu, A. Shahroudy, D. Xu, and G. Wang, "Spatio-temporal lstm with trust gates for 3d human action recognition," in *European Conference on Computer Vision*. Springer, 2016, pp. 816–833.

[24] F. A. Gers, N. N. Schraudolph, and J. Schmidhuber, "Learning precise timing with lstm recurrent networks," *Journal of machine learning research*, vol. 3, no. Aug, pp. 115–143, 2002.

[25] A. Shahroudy, J. Liu, T.-T. Ng, and G. Wang, "Ntu rgb+d: A large scale dataset for 3d human activity analysis," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.