Hands on Wheel Classification Based on Depth Images and Neural Networks

Jan-Christoph Schmitz[a], Stephan Tilgner, Kathrin Kalischewski, Daniel Wagner and Anton Kummert

University of Wuppertal, School of Electrical, Information and Media Engineering, 42119 Wuppertal, Germany

Abstract. This paper describes a system to automatically observe if the driver has his hands on the wheel, which is important to know that he can intervene if necessary. To accomplish this an artificial neural network is used, which utilizes depth information captured by a camera in the roof module of the car. This means that the driver and the steering wheel are viewed from above. The created classification system is described. It is designed to require as little computational effort as possible, since the target application is on an embedded system in the car. A dataset is presented and the effect of a class imbalance that is incorporated in it is studied. Furthermore, it is examined which part, i.e. the depth or the intensity image, of the available data is important to achieve the best possible performance. Finally, by examining a learning curve, an experiment is made to find out whether the recording of further training data would be reasonable.

1 Introduction

As car automation progresses in the direction of autonomous driving, the driver is supported ever further. In an increasing number of situations cars can drive autonomously, which most probably will develop in a smooth transition to a point where the driver only has to intervene in certain rare situations, and ultimately no longer at all.

Especially in this transition period it is important that the vehicle's control unit has information about the driver's condition and activities, as it is particularly difficult for people to be alert over a longer period of time if they are not strongly challenged [1]. Knowing whether the driver has his hands on the steering wheel can be used in situations where the driver's attention is required. For example, autonomous driving should not be switched off if the driver cannot take control.

This work deals with the processing of image data from a depth camera, which is directed into the interior of the vehicle and is located in the roof module, using artificial neural networks. The aim is to find out whether it is possible to use this technique to classify if the driver has his hands on the steering wheel or not and to build a working system if possible. This includes finding out what part of the available data is needed for good performance. Due to the target application on an embedded system in the car, which limits the available computing power, rather small neural networks need to be used.

2 Related work

A lot of work is devoted to monitoring the driver's condition in order to further improve road safety or the user experience.

In one fundamental study, possibilities are considered to improve vehicle safety by means of image processing techniques [2]. For this purpose, both the environment as well as the condition of the vehicle and the driver himself are taken into account. Therefore, for example, the occupant position and posture, the intentions of the driver and the surrounding, e.g. the lanes and obstacles such as other vehicles, are analysed. Since the observation of the driver is an important aspect, [3] provides an overview of various studies dealing with the evaluation of the driver's condition, his behavior and the prediction of his intentions. In [4] this overview is extended by more recent research work, although the listed collection does not concentrate exclusively on the driver himself, but also contains studies which consider people outside the car.

Due to the low costs and the possibility to use depth image data, many studies rely on the *Microsoft Kinect* camera. A comparison of different machine learning methods concludes that artificial neural networks have advantages over other methods when it comes to identifying activities, such as using a mobile phone, on the basis of data obtained from the Kinect [5].

As a further research shows, the hands are a crucial feature to identify human activities [6]. In their study, only image sections around the hands are passed to a recurrent neural network which performs the evaluation.

In [7] the behavior of the driver is analysed on the basis of his hand activities, which includes the interaction with

[a] Corresponding author: jaschmitz@uni-wuppertal.

the steering wheel. Color and depth images are used, which are divided into different regions, such as the steering wheel or the gear lever. A Support Vector Machine distinguishes the different activities by recognizing in which of these regions the driver's hands are active. Occlusions of the hands are a serious problem, which is addressed in [8].

[9] uses the tool *XMOB*, which is presented in [10], to get the hand positions. If these are in a previously marked region, it is decided that the particular hand is on the steering wheel.

Of particular interest is [11]. Besides the question if the driver uses a mobile phone, the number of hands on the steering wheel is determined by artificial neural networks. Bounding boxes for the hands and the steering wheel are detected in a color image. If the intersection area between the bounding box of a hand and that of the steering wheel is greater than 5 % of the size of the hand bounding box, then the hand is classified as on the steering wheel. The detection method used to accomplish this is based on *Faster R-CNN* [12]. Its disadvantage, to not recognize very small objects, such as the hands in this case, is overcome by training the detection network with different scales of the Faster R-CNN's region proposals. Weaknesses of this method are, for example, the high computational effort required, and errors occurring under difficult lighting conditions.

Another approach uses crops of the hands to find out whether the shown hand grabs the steering wheel or a mobile phone [13]. To obtain the hand crops, a hand localization is performed using a CNN that extracts regions where a hand may be located. The next step is a pixel-wise skin classification to segment the hand so that a selection can be made from these regions. The *Histogram of Oriented Gradients* (HOG, [14]) method is used to extract important properties of the hand from these segmentations. Finally, an SVM uses these properties to classify whether the driver grabs one of the objects. All in all, this approach results in a very high computational effort, especially due to the large CNN, which is used for the hand localization.

Datasets developed especially for hand detection in cars are important for continuously improving algorithms and comparing them with each other. In [15] a dataset is presented which tries to represent driving scenarios that are as realistic as possible. It contains color images taken from different angles and with different illumination intensities. The annotations include bounding boxes for each hand as well as information on whether it is the left or right hand and whether it belongs to the driver or co-driver. In addition, there is information about the number of hands on the steering wheel.

Beside a neural network that detects hands or objects held by the driver, [16] introduces a simple procedure for semiautomatic annotation of hands. For this purpose, the driver wears green gloves and red wristbands. Chroma keying is used to detect the hand positions based on these colors. The data obtained in this way can then be used, for example, to train a hand detection method based on depth image data.

3 Hands on wheel classification system

3.1. System setup

The basis of the hands on wheel classification system is a *Time of Flight* (ToF) camera, which is integrated into the roof module of the car, so that the driver is seen from above, and provides intensity and depth information about the front part of the driver's cabin. This information is represented as images with a QVGA resolution. An example of these images is shown in Figure 1. Through an active lighting the images are independent of the ambient light. The intensity image Figure 1a results from measuring the reflected amount of this active lighting. Objects that poorly reflect this light are causing noise in the depth image Figure 1b.

(a) Intensity image (b) Depth image

Figure 1. Example of the information captured by the ToF camera. The depth image contains high noise in some areas.

The images are recorded at a frame rate of 30 Hz, but only every third image is used to prevent the dataset from containing too many similar examples.

Only the left half of the images is used for the classification, since the other one contains no relevant information and the computational cost is lower the smaller the input image is.

The resulting dataset is saved as a multidimensional array. The first dimension represents the different examples, the second and third dimension contain the individual pixels and the fourth dimension distinguishes the intensity and depth information.

3.2 Network architecture

The network architecture of the hands on wheel classifier is depicted in Figure 2. The first three stages extract relevant features of the input images. Each of them contains a convolutional layer with size 3×3, which is followed by a ReLU activation function. Max pooling is applied during the first two stages to reduce the spatial size and thus the number of parameters in the fully connected layers. Dropout [17] is applied after the third stage to prevent overfitting. After that the extracted features are evaluated by two fully connected layers. The first one is followed by a ReLU activation function, whereas the second one is followed by a softmax activation function to predict a probability for each class.

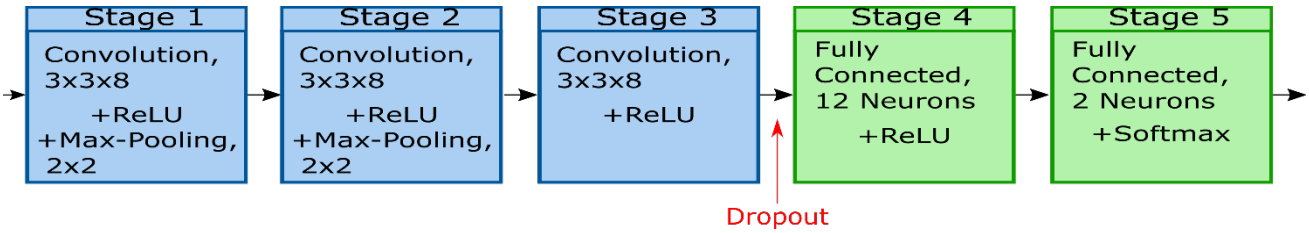

Figure 2. Architecture of the hands on wheel classifier. It consists of three convolutional layers, followed by two fully connected layers. Dropout [17] is applied between stage three and four to avoid overfitting. The last layer is followed by a softmax activation function to obtain probabilities for the occurrence of the classes.

3.3 Training procedure

The network is trained for 100 epochs, which consist of several mini-batches. Before the first epoch starts the weights are initialised to small random values based on a Gaussian distribution [18]. To calculate the difference between the predictions and labels, the cross-entropy loss function

$$H(y, \hat{y}) = -\sum_{k=1}^{K} y_k \log \hat{y}_k \qquad (1)$$

is used. $\hat{y}_i$ is the prediction of the classifier, y_i the value of the label and K is the number of different classes. For every mini-batch the average loss over all contained examples is calculated. *Weight decay* is added to this loss as a regularizer. The Adam optimizer [19] is used to update the weights dependent on the loss after each training step.

Several hyperparameters result from this architecture. These are the size of the mini-batches, the used amount of dropout and the amount of weight decay.

4 Experimental evaluation

4.1. Dataset

To evaluate the classifier, a dataset was created which contains a lot of variation to represent as many real situations as possible. For this purpose several video streams were recorded that contain different cars, several positions of the driver's seat and steering wheel, and varying drivers with diverse clothing. Figure 3 shows three different examples, in which the driver is wearing different clothes. Additionally two different versions of the ToF camera were used. In each stream only one class is executed, which simplifies the labelling procedure.

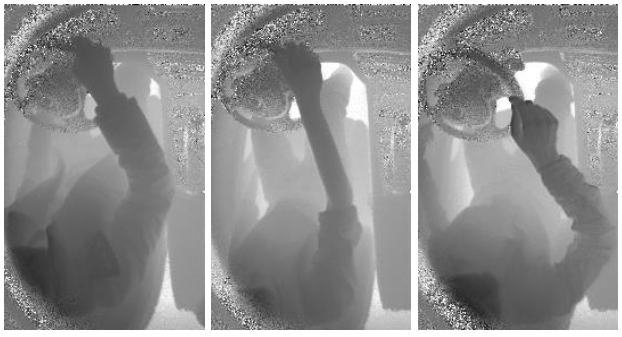

Figure 3. Examples for the variation in appearance of similar body postures caused by different clothes and body shapes.

Table I shows the composition of the dataset. The rows distinguish different experimental setups, for which the total number of images and the corresponding proportions are shown.

Table 1. Composition of the used dataset. Different experimental setups and the amount of corresponding images are shown in different rows. The dataset contains images from different versions of the ToF camera in car 1.

Setup	Proportion	Number of images
Car 1, Camera 1	65.12 %	58800
Car 1, Camera 2	3.99 %	3600
Car 2, Camera 1	30.89 %	27900
Total	100 %	90300

The distribution of the classes is depicted in Table II. It is recognizable that the dataset contains more examples in which the hands of the driver are not at the steering wheel.

Table 2. Distribution over the classes in the dataset. Only in about one third of the data the driver has his hands on the steering wheel.

Setup	Hands on wheel	Hands off wheel
Car 1, Camera 1	36.73 %	63.27 %
Car 1, Camera 2	50 %	50 %
Car 2, Camera 1	22.58 %	77.42
Total	32.89 %	67.11 %

To reasonably estimate the accuracy of the model, 5-fold cross-validation is carried out. In each run of the cross-validation, different 20 % of the examples are used in a separate validation set, which is not used during the training process. The presented results in this paper are the mean over the runs of the cross-validation. This procedure is intended to cover the generalization to all cases that occur in the dataset.

It is to mention that the dataset contains a lot of challenging cases, in which for example the driver is pretending to grasp the steering wheel. Therefore, the error on the dataset is expected to be higher than in a realistic scenario.

4.2. Analysis of the class imbalance

When a threshold is used to get a binary prediction from the classifier, the result is either right or wrong. Because of the class imbalance in the dataset (Table II) a simple threshold of 0.5 is not sufficient. Hence, the performance metric used in the evaluation is the area under

the Precision-Recall Curve (PR AUC), which has a maximal value of 1.

To figure out how the class imbalance is effecting the model, the false negative rate, false positive rate and total error are plotted against the threshold in Figure 4. The total error is the proportion of misclassified examples. At a threshold of approximately 0.2 the false negative and false positive rate are the same. This means that the classifier is better in detecting the class in which the driver is not grasping the steering wheel. The minimum total error is approximately 5.4 %. Thus, at a threshold of 0.2, with a value of about 6 %, the total error is not far from the minimum, which means that the change of the threshold can counter the effect of the class imbalance in the dataset.

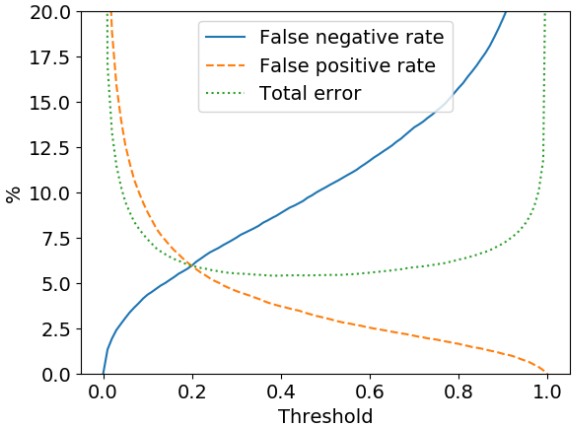

Figure 4. Classification results with different thresholds. The intersection point is at a threshold of approximately 0.2 and the total error at this point is not much higher than the minimum total error.

4.3. Input analysis

In order to obtain an efficient classification system, it is important to know which of the possible inputs contribute to the result. Therefore, it is investigated to what extent the intensity and depth images contain complementary information for the hands on wheel classification.

Figure 5 shows a comparison of the PR AUC for the classifier when different inputs are used. The different inputs are the depth image, the intensity image and both images, where the images are stacked in the fourth dimension. When the input is the intensity image alone, the results of the classifier are significantly worse than the results achieved by using the other two input methods. Since using both images as input does not induce an improvement over using the depth image alone, it can be concluded that no relevant additional information is obtained from the intensity image. Therefore, the depth image as input is sufficient and used for the further experiments.

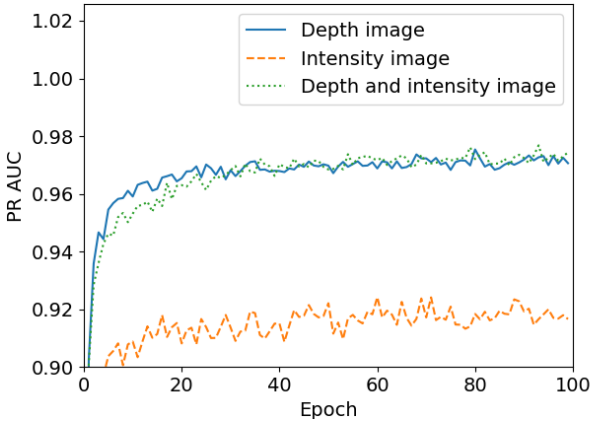

Figure 5. Comparison of the classifier using the depth image, the intensity image and both images as input. Using only the intensity image, a significantly worse result is achieved. There is no improvement over training on the depth image alone if the classifier is trained on both images.

4.4. Examining the learning curve

In order to find out whether an increase in the amount of training data would lead to a better result, the learning curve was examined. This is interesting, because the recording of new training data is a great effort and should therefore only be performed if an improvement in performance is likely. However, this method does not provide any information on how the result will behave if the new data covers certain cases that do not yet exist in the training data. This means that it can still be useful to generate new training data for cases where the system encounters many errors. In order to obtain the learning curve, the artificial neural net is trained multiple times. Initially, only a small proportion of the available training data is used. This percentage is increased in each training run so that finally all data is used. The result is a curve in which the error depends on the amount of training data. If this has already converged during the last training runs, it is only worth recording more data if it covers specific cases in which the performance is unsatisfactory. For each training run, the best value occurring during the training is used in the learning curve.

With this procedure the curves shown in Figure 6 are produced. During the generation of these curves, 20 % of the data set was used as a fixed test set. It is interesting to note that the result on the training data, after a certain point, remains constant as the size of the data set increases. This suggests that the network architecture of the classifier does not allow memorization. Otherwise, the result would become worse with a larger amount of training data, because more and more would have to be learned by heart.

The test data set shows a strong increase in performance at the beginning. After a slump at about 65 % of the training set, there is another positive trend towards the end. This means that a larger amount of training data could be helpful for the classifier to generalize to the used test data.

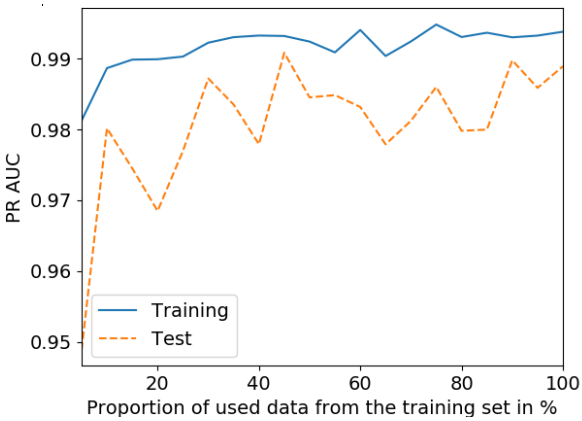

Figure 6. Learning curves of the hands on wheel classifier. A larger portion of the training data is used for each training run. With a larger number of training examples the result improves especially at the beginning. At the end a positive trend on the test data can be seen again.

5 Conclusion

In this paper it is investigated to which extent the information obtained from the images of a ToF camera as the input for an artificial neural network can be used to detect whether the driver's hands are on the steering wheel. This camera observes the interior of the vehicle and is mounted in the roof module. The resulting system performs well on a challenging dataset, which means that better results can be expected in practice. In addition the frame-wise predictions can be filtered to reduce the effect of outliers. However, the remaining error is still a little too high to completely rely on the classifier for the safety-relevant question of whether the driver has his hands on the steering wheel. The remaining errors occur in situations where the driver's hand is above his thigh or when the left arm is not in the image while the left hand touches the steering wheel in the outermost part of the image. To avoid the latter error, a larger field of view would be necessary. But these remaining errors are false negatives rather than false positives. This means that the system does not often mistakenly think the driver is ready to drive.

An analysis of the effect that arises from an imbalance in the dataset demonstrates that the effect of this imbalance can be reduced by choosing an appropriate threshold for the binary classification. Since it is more important to detect if the hands are off the wheel, this threshold could be used as well to get a better prediction on the corresponding class while accepting worse results on the other class. An alternative to adjusting the threshold would be balancing the dataset by generating more examples of the smaller class or weighting the loss function according to the class ratio.

Additionally, it is shown that the use of the depth image is sufficient to obtain a good classification result with the used system, as an additional use of the intensity image does not lead to any improvement.

Furthermore, an examination of the learning curve showed that an extension of the data set could lead to better performance.

For the future, the differentiation of more classes, which are combined afterwards, could have a positive effect. For example, recognizing a blocked view could lead to greater stability.

Acknowledgment

Special thanks goes to Aptiv for making this work possible by providing the experimental setup and constant support. In particular we thank David Schiebener and Alexander Barth for helpful discussions and feedback.

References

1. M. Kyriakidis, J. C. F. Winter, N. Stanton, T. Bellet, B. Arem, K. Brookhuis, M. H. Martens, K. Bengler, J. Andersson, N. Merat, N. Reed, M. Flament, M. Hagenzieker and R. Happee, TIES **53**, 1-27 (2017)
2. M. M. Trivedi, T. Gandhi and J. McCall, TITS **8**, 108-120 (2007)
3. C. Tran and M. M. Trivedi, *Visual Analysis of Humans*, 597-614 (2011)
4. E. Ohn-Bar and M. M. Trivedi, TITS **1**, 90-104 (2016)
5. Y. Xing, C. Lv, Z. Zhang, H. Wang, X. Na, D. Cao, E. Velenis and F.-Y. Wang, TCSS **5**, 95-108 (2018)
6. F. Baradel, C. Wolf and J. Mille, ICCVW, 604-613 (2017)
7. E. Ohn-Bar and M. Trivedi, IV, 1034-1039 (2013)
8. A. Rangesh, E. Ohn-Bar and M. M. Trivedi, CVPRW, 1224-1231 (2016)
9. C. Tran and M. M. Trivedi, ICVES, 97-101 (2009)
10. C. Tran and M. M. Trivedi, ISM, 446-447 (2009)
11. T. H. N. Le, Y. Zheng, C. Zhu, K. Luu and M. Savvides, CVPRW, 46-53 (2016)
12. S. Ren, K. He, R. Girshick and J. Sun, TPAMI **39**, 1137-1149 (2017)
13. Siddharth, A. Rangesh, E. Ohn-Bar and M. M. Trivedi, ITSC, 2545-2550 (2016)
14. N. Dalal and B. Triggs, CVPR, 886-893 (2005)
15. N. Das, E. Ohn-Bar and M. M. Trivedi, ITSC, 2953-2958 (2015)
16. A. Rangesh and M. M. Trivedi, CVPRW (2018)
17. N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever and R. Salakhutdinov, JMLR **15**, 1929-1958 (2014)
18. X. Glorot and Y. Bengio, PMLR **9**, 249-256 (2010)
19. D. P. Kingma and J. Ba, ICLR (2015)