In-cabin Activity Taxonomy using Body Pose Features

Nikhil Karnwal
University of California San Diego
La Jolla, California, United States
nkarnwal@ucsd.edu

Rajeev Dixit
University of California San Diego
La Jolla, California, United States
rcdixit@ucsd.edu

Seth Farrell
University of California San Diego
La Jolla, California, United States
swfarrel@ucsd.edu

Leeor Nehardea
University of California San Diego
La Jolla, California, United States
lneharde@ucsd.edu

Abstract

In this paper, we evaluate the impact of extracting intermediate features related to the driver and presence of different objects on the accuracy of predicting in-cabin activities in the context of autonomous vehicles. We restrict our study to a single camera setup, which provides 2D images which look at the driver of a moving vehicle. We analyze whether extracting driver related features such as head pose and body pose and detecting other objects such as cellphone, cup or the steering wheel will give as better accuracy in classifying in-cabin activities in comparison to a deep convolutional neural network, without any added information. All the figures are reported on the Statefarm Distracted Driver Dataset, which consists of a single Rgb image taken from the passenger perspective, and represents ten different activity classes. To that effect, we also analyze different SOTA methods for face and body pose detection, and in-cabin object detection and select those best suited to our dataset. We then combine the features obtained from these methods and pass it through our vanilla neural network and compare our performance to two CNN baselines of AlexNet and Vgg-Net.

1. Introduction

In the past few years, there has been tremendous progress on autonomous vehicles research, and we have seen the launch of vehicles with upto level 3 and 4 of autonomy as described by Society of Automotive Engineers [8]. Lately, some accidents have been reports in such vehicles, and hence it is important to analyze driver behavior for safety and well-being of all stake-holders, until we have vehicles with level 5 of autonomy, where the presence of human drivers is optional. Hence, classifying in-cabin activity is a very important problems in autonomous vehicles. While there has been a lot of research in predicting driver take-over readiness, for example [6], our paper focuses on accurately classifying different in-cabin activities.

Figure 1. Example of In-Cabin view

For our purpose, we decided on the Statefarm Distracted Driver Dataset [5], which was released as a part of a Kaggle competetion "Distracted Driver Detection" in 2016. The dataset 1 has ten classes, where c0 stands for normal driving, and c1 - c9 consist of common secondary activities such as talking or texting on the phone, or drinking a beverage, as shown in Table 1. There are a total of 22,424 images, and they are evenly distributed across all the classes. The Kaggle competition is relatively old, and with five years of advances, our primary objective was not compete on the leaderboard, but we could use it validate if our approach by seeing if it achieves to a good score. Based on our day-to-day experience, we hypothesized that several key features might be helpful to classify the driver's action. These key features are the driver's head and body pose, and the driver's hand location especially in context of objects like the steering wheel, cell-phone or cup. For instance, when at least one hand is away from the steering wheel, and the driver's body is bent forward and looking downward, the driver is

likely texting. Therefore, in our approach, we first obtain a few important features by running the input image through our first component. We then process these features and add them as an input to our second component, which is a vanilla neural network that will output the predicted class. We compare this approach with two baselines of AlexNet [13] and Vgg-Net [21] and with two other papers [3] [1] which have used variations of deep neural networks on the same dataset.
It is important to note that we also attempted to detect the steering wheel by using a unique approach of circle detection and ellipse detection using Hough Transform, but ultimately failed to do so due to the unique perspective in the images of the dataset and large occlusion in many images due to the driver's hands, etc.

Class label	Driver activity
c0	Safe driving
c1	Texting - right hand
c2	Talking on the phone - right hand
c3	Texting - left hand
c4	Talking on the phone - left hand
c5	Operating the radio
c6	Drinking
c7	Reaching behind
c8	Hair & makeup
c9	Talking to the passenger

Table 1. Classes for in-cabin activities.

2. Related Research

In this section, we summarize previous literature on In-cabin driver activity recognition, by dividing into five key components: General work on In-cabin activity recognition, Driver head pose estimation, Driver hand and body pose estimation, In-cabin object detection for common objects like cell-phone, cup or water bottle and steering wheel, and previous work on Statefarm Distracted driver dataset.

2.1. In-cabin Activities Recognition

In one paper [15], the researchers designed a robust architecture which extracted features from the cabin image and a CNN to classify in-cabin activity. Their approach used fewer parameters than state-of-the-art models and outperformed most other models proposed thus far at the time. They also suggested that additional features such as posture of the driver may significantly improve performance of the model.

In another paper [11], a very similar approach to ours was used. This included a transfer learning approach in which various pre-trained models were used and then fine-tuned to improve their accuracy. Some of the models used include VGG16, ResNet50, and Mobile Net.

Lastly, in [19] identified three key requirements for a practical real-time detector for in-cabin detection. These being high accuracy, high speed, and a low number of parameters. To this end they proposed an architecture which was based on a decreasing filter size. Testing their model on the state farm dataset resulted in an extremely high accuracy of 99.87 percent.

2.2. Driver Head Pose Estimation

There has been a lot of prior research in estimating driver head pose, which includes image processing and traditional ML approaches [17], Synchronised multi-modal data capture [22] and Point-Cloud based approach using depth cameras [7]. While these methods give impressive results, they are not suited to our dataset due its lack of depth information, and having a single camera which has a different perspective than what is more common for a specific head pose detection task. However, capturing head pose information was important for our problem because it can help distinguish between activities like c0 - Safe driving, c7 - Reaching behind, c9 - Talking to the passenger and c5 - Operating the radio. Hence we looked into other head-pose detection approaches which may not be restricted to in cabin activities. The FSA-Net by Tsun-Yi Yang, et al [23] is one such approach, which is near State-of-the-art, that we looked at, but its disadvantage for us was that it required an additional step of extracting only the face from the In-cabin image present in our dataset. With this we finally arrived at Img2pose [2] by Vitor Albiero, et al which was end-to-end i.e it detected face by itself, provided state-of-the-art performance on face detection benchmarks and performed well on poses with wide angles.

2.3. In-cabin Object Detection

Our dataset is unique, mainly due to the image point of view. There are several papers we found that focus on classifying cell phone usage [14], for example. Yet, in our task, we also need to determine which hand holds the phone and if the driver is texting or talking and there is no previous work which makes this kind of distinction.
Also, several papers attempt to find wheels in context of exterior view of a car, such as [18]. However, we could not find any research paper that attempts to find the steering wheel, especially not from such an angle. After analyzing the steering wheel from the perspective in our image, we looked image processing approaches to find ellipses, primarily [9]. However this approach doesn't yield significant results maybe due to partial occlusion due to hands, having other objects in close proximity and the steering color closely matching the background.

2.4. Driver Hand / Body Pose Estimation

There exist various previous approaches centered around extracting the driver hand position, especially in the context of a steering or a cell phone. T. Hoang Ngan Le, et al from CMU have used a Multiple Scale Faster-RCNN [14] for predicting if the hands are on the steering wheel or using a cell-phone by using images from different perspectives such as forward view, face view and lap and hand view. Akshay Rangesh, et al from UCSD's LISA lab detect a wider set of objects held in the hands including smartphones, tablet, drinks or books in addition to steering wheel in HandyNet [20] by using infrared sensors in addition to a RGB image. However, both of these methods couldn't be used in our dataset due to difference in perspective and availability of a single RGB image.
Hence we shifted our attention to body pose detection, which would also be more useful for our dataset to predict activities like c7 - Reaching behind and c9 - Talking to the passenger. Manuel Martin, et al present an algorithm to detect 3D driver upper body pose using a single depth camera in [16], and the perspective is from passenger side, which is similar to our dataset, but the images in our dataset do not capture depth information. We explored another model, HigherHRNet [4], which uses RGB image to extract impart body key points and compute pose for it. Since it is trained irrespective of camera view, it perform very well on side view images.

2.5. Statefarm Distracted Driver Dataset

The Statefarm dataset was published on Kaggle in 2016, and thus there are a some papers which report work on this dataset too. [1] uses AlexNet and Inception-V3 along with Hand and Face localization, whereas [3] used Vgg-Net and also proposes their modification to it to run more efficiently on the dataset.

3. Methodology

3.1. Baseline

To compare our approaches against a good starting method, we decided to use the AlexNet [13] Convolutional Neural Network as our baseline method as it had been done in [1]. We modified the AlexNet slightly to predict only 10 classes in our dataset as opposed to 1000 used in the original paper for the ImageNet dataset. We used a pre-trained model, which was trained on the ImageNet dataset. The Alexnet baseline performed well on the test data which had known subjects but not too well on unseen subjects. Hence we decided to include another Baseline of a newer and better CNN architecture which is why we went with VGG-16 [21], which was also used in [3]. We modified the VGG-Net similar to the AlexNet, and it was also pre-trained on the ImageNet dataset. We trained both the baselines on 30 epochs, as we were qw

3.2. Face Pose Estimation

We decided to use Img2pose [2] for obtaining the head pose of the driver. This is an end-to-end model which also localizes the face in the image in addition to estimating the head pose. Since our dataset doesn't have labels for head pose or head location, we couldn't have trained it on it. Hence we pre-trained the model on the AFLW-2000 dataset [12] since it has faces with wide range of angles which will suit the perspective of the images in our dataset. We extracted the face pose in a offline way for all images at once, and stored them in a json file to be consumed by our model later. This ensured that the head pose features were the same across different experiments. We obtained face pose in 6 degrees of freedom which included angles for Yaw, Pitch & Roll and 3D coordinates for face translation. Along with that we also extracted the coordinates for the face bounding box, to be used as features in our model, as shown in the figure below. Face location, when combined with locations of cell-phone or water bottle can be useful for predicting activities like talking on the cell phone or drinking a beverage.

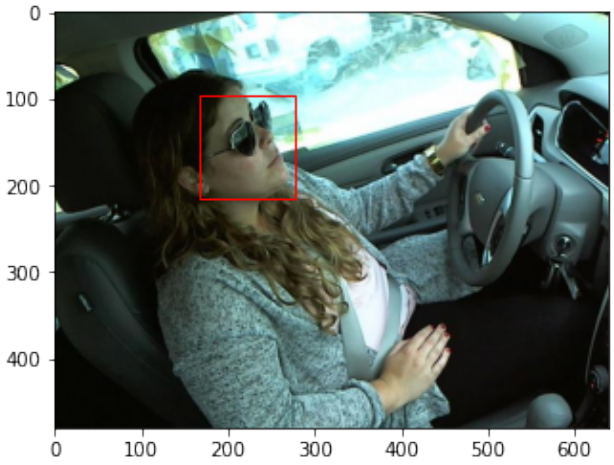

Figure 3. Face location detected using Img2Pose

3.3. Body Pose Estimation

Body pose is highly correlated to an activity that a person perform and crucial in distinguishing different activities. This led us to use Body pose of the driver as significant features in addition to other features. We exploited HigherHRNet [4] model for extracting 17 important key points as shown in the following table.

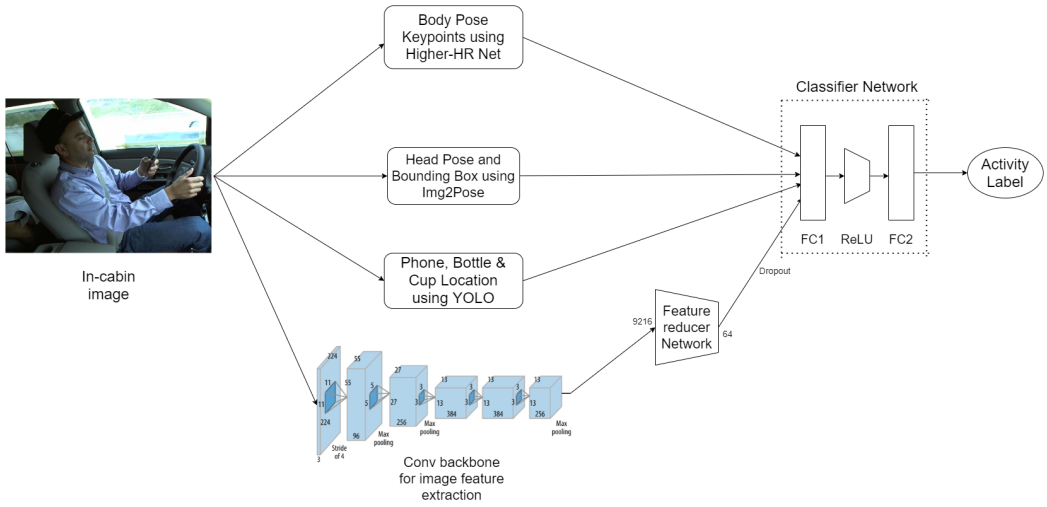

Figure 2. Combined Network Architecture.

Key Points - Body Parts
Nose
Eye left-right
Ear left-right
Shoulder left-right
Elbow left-right
Wrist left-right
Hip left-right
Knee left-right
Ankle left-right

Table 2. Body parts covered through key-points

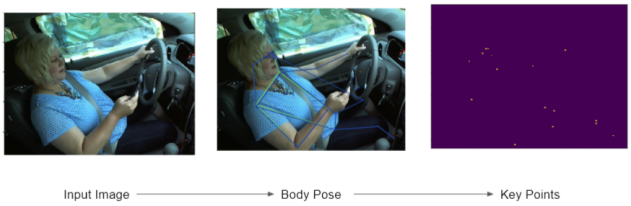

Figure 4. Body pose and Key points using HigherHRNet

Above figure 4 show an example of a driver performing an activity and how model accurately predict upper body pose with corresponding key points.

3.4. Object Detection

We wanted to use in-cabin object detection to increase the model accuracy. The initial idea was to detect the distance between the wrists and the steering wheel. If both wrists are very close to the steering wheel, we can assume with a high probability that both hands hold the steering wheel. If that is the case, then we have created a strong feature to help to detect a class (c0 - "normal driving"). However, we could not find a sufficient way to detect the steering wheel in the image. One of the main issues is that the steering wheel seems like an ellipse because of the image viewpoint, which is harder, and more computationally expensive to find (compare to a circle). Next, we decided to try and detect the steering wheel and other relevant objects, such as cell phones and bottles using YOLO [10]. We chose to use YOLO because of its simplicity to, use and its performance. Our first step was to find datasets with the relevant classes. We used Google Open Dataset V6. We downloaded 2,000 images from each class and used transfer learning with YOLO to detect these images. Unfourtently, the detection performance on these classes was poor. Thus, we finally decided to use YOLO with its original weights. YOLO was originally trained on the COCO dataset. We found out that there are 3 objects in this dataset that are useful for us - bottles, cups, cell phones. These objects are expected to be detected in classes c1: texting - right, c2: talking on the phone - right, c3: texting - left, c4: talking on the phone - left, and c6: drinking. Hence, our last step in the object detection area was to run YOLO over all the train images, collect its prediction, and if it predicted a relevant class (bottle/ cups/ cell phone) get its bounding box location. This will be used later in the network as an additional feature.

3.5. Classifier

After having extracted driver related features such as body and head pose and localizing in cabin objects such as cell-phone and cups, we come two our second stage which is the classifier network as shown in Figure 1. The features were extracted beforehand and stored in json files, so we

used them in an offline way while training our classifier network. In addition to the extracted features, we also ran experiments which included outputs from Convolutional layers from a CNN backbone, as a image feature map representation to complement our extracted features. We used AlexNet as the CNN backbone, but it can be easily substituted with any CNN architecture. Our classifier consists of two fully connected layers and a ReLU layer in between. The CNN backbone we used was pre-trained on the ImageNet dataset, and hence we froze its weights in the training process. The classifier was trained on different experiments, and trained to 30 epochs, as achieved saturation in the validation loss.

4. Experimental Evaluation

4.1. Dataset Preparation

We used the Statefarm Distracted Driver Dataset which is a public dataset available on Kaggle, and was part of a Kaggle competetion in 2016. We work on the 22K training images available in the dataset, by performing our own split into training, testing and validation datasets. Based on the performance of the models on different splits, we report our results on two different methods of data preparation. In the first method, we perform a simple split on our data in the ratio 80:10:10 for training, testing and validation using random distribution similar to what is done in [3].
After seeing a large gap between the performance on our test data, and the kaggle testing images for the leaderboard, we decided to test on a different preparation method. Here we ensured that the subjects in our test and validation data are not seen in the training phase. Out of the 26 unique subjects available in training, we approximately used 80% of them for training and 10% each for validation and testing. This method is a good test of whether the model can generalize beyond the subjects seen in training, and a much more robust model will be required to perform well on this data.

4.2. Metrics for Evaluation

Our objective is to correctly classify driver activity into the ten given classes. Hence we present our results on two metrics; Percentage accuracy of prediction and multi-class log loss. They are defined as follows:

$$Accuracy = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}} \quad (1)$$

$$Logloss = \frac{-1}{N}\sum_{i=1}^{N}\sum_{j=1}^{K} y_{ij} log(p_{ij}) \quad (2)$$

4.3. Qualitative Results

Below are some images showing the results of running our baseline models on our test data. We show the ground truth labels of randomly selected images as well as the labels predicted by our models. Our baseline models were trained on 10 epochs as we did not see much improvement after that. More epochs were use later in testing our hyper parameters the results of which are shown later in this paper.

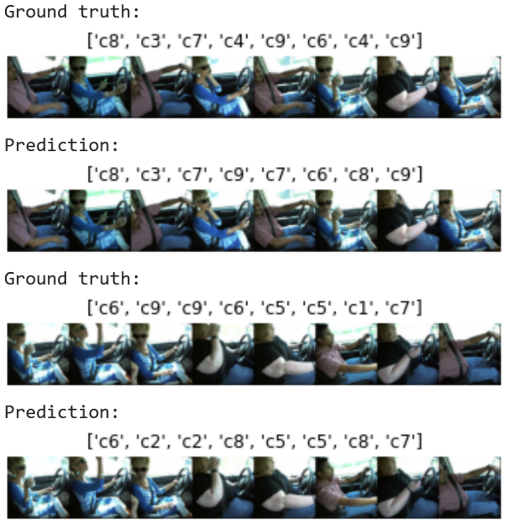

Figure 5. Alexnet Baseline Predicted Labels

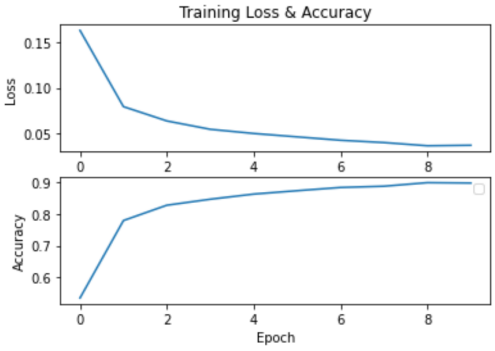

Figure 6. Alexnet Baseline Training Loss Accuracy

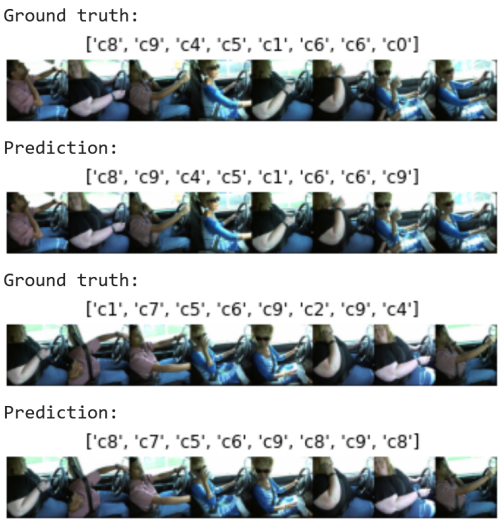

Figure 7. VGG16 Baseline Predicted Labels

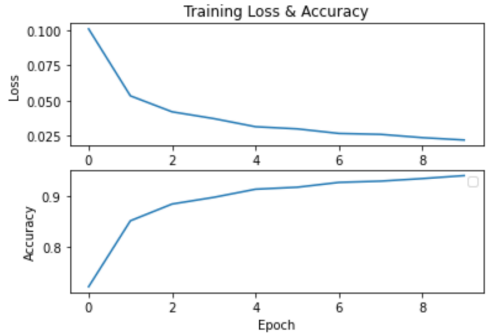

Figure 8. VGG16 Baseline Training Loss Accuracy

4.4. Quantitative Results

In this section, we present Confusion matrix for our models, loss and average accuracy for random data set and person wise split data set.

Model	Train Acc.	Val Acc.
Alexnet	88.15	99.04
Body pose	98.28	96.72
Features + Head pose	96.93	98.67
Features + Body pose	98.51	99.13
Features + Body pose + Head pose	**98.88**	**99.19**

Table 3. Results on Random Data set

Above results shows accuracy of each model on random data set. We can see that by using only Body pose coordinates along with simple FCN, our model trained very well and gave 98.28% and 96.72% on train and validation data set respectively. Over all, combining body pose, head pose and feature map gave best results. We did not use test data set here as all models are giving above 95% accuracy so we didn't put effort in hyper parameters tuning.

Now we have provided results on person wise split data set where train , validation and test data does not have common participant.

We experimented with some of the hyper-parameters including learning rate and weight decay. These were all obtained using the vgg16 model and training for 30 epochs.

Learning Rate	Weight Decay	Train Acc.	Test Acc.
0.05	0.0005	0.81	0.71
0.01	0.005	0.85	0.76
0.01	0.001	0.91	0.81
0.01	0.0005	0.90	0.73
0.005	0.005	0.92	**0.84**
0.005	0.001	0.94	0.82
0.005	0.0005	0.92	0.72

Table 4. Hyper Parameter Testing Results

Using above table, we decided to use learning rate = 0.005 and weight decay = 0.005 for training our baseline models.

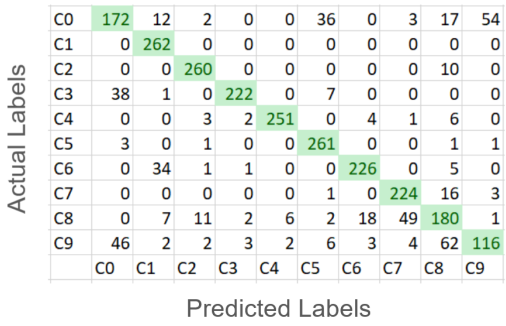

Figure 9. Alexnet Baseline Confusion Matrix

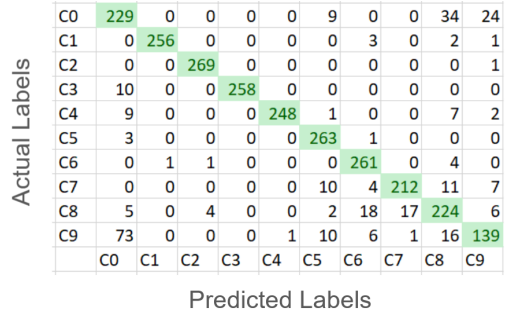

Figure 10. VGG16 Baseline Confusion Matrix

Figure 9 and 10 confusion matrix for baseline models on test data set.

Model	Train Acc.	Val Acc.	Test Acc.
VGG	92.40	86.4	84.42
Body pose	91.95	75	65
Feat. + Head pose	-	-	-
Feat. + Body pose	-	-	-
Feat. + Body + Head	-	-	-

Table 5. Results on Person wise split data set

Above table shows results of our models using best hyper parameters on person-wise split data set (We could not do parameter tuning for all so results are still pending). We can see that VGG is giving 84.42% on test data set while body pose has 65%. This is contrary to what we saw in Random data set where Body pose has performance close to other models. We looked into the cases where miss and found following results as can be seen from confusion matrix 10.

- Model is not able to distinguish between driving safely or using radio, or talking to passenger
- Texting using right hand or drinking
- Texting using left hand or talking on mobile
- Using radio or texting.

It is evident that body pose for texting or drinking might be similar but object is different. So, we decided to use Yolo as object detector and hope to get better performance. After using Yolo, we found out that it is detecting objects only for 60% of images as objects are very small in given image and occluded by palm so we dropped it for now. We have planned to work on it later by increasing the resolution of images.

5. Concluding Remarks

Based on the randomly generated data split, out model performed marginally better than the Alexnet baseline. However since, the the data split is not challenging enough for the baseline, the real extent of our improvement could not be indisputably evaluated. Another area for further evaluation is that, whether our method works irrespective of the backbone CNN ? For example if I use a Resnet baseline and change the backbone of our network to Resnet, does it offer equally good improvement in accuracy. One more area for futher work is generalizablilty of our model; we observed that the validation accuracy drop substantially on using the second method of data preparation where unknown subjects are fed to the model after training, we could definitely improve on that.
This paper gives a different approach on driver activity classification which unifies different feature extraction methods instead of just going deeper, and gave encouraging results in general. Some more effort is needed to reproduce these results maybe using different back-bone or on a different dataset, and confirm its efficacy. Another interesting idea proposed in our paper was detecting steering wheel by using an ellipse detector for passenger side perspective images and more work and refining this approach could give good results in this regard.
One more area for further work is to evaluate whether all the different features can be extracted in parallel and in minimal time to make this end-to-end method real-time, by getting frame rate of over 20-30 Fps.

References

[1] Yehya Abouelnaga, Hesham M Eraqi, and Mohamed N Moustafa. Real-time distracted driver posture classification. *Neural Information Processing Systems Workshop on Machine Learning for Intelligent Transportation Systems*, 2018. 2, 3

[2] Vitor Albiero, Xingyu Chen, Xi Yin, Guan Pang, and Tal Hassner. img2pose: Face alignment and detection via 6dof, face pose estimation. *IEEE Conf. Comput. Vis. Pattern Recog.*, 2021. 2, 3

[3] Bhakti Baheti, Suhas Gajre, and Sanjay Talbar. Detection of distracted driver using convolutional neural network. *IEEE Conf. Comput. Vis. Pattern Recog. Worksh.*, 2018. 2, 3, 5

[4] Bowen Cheng, Bin Xiao, Jingdong Wang, Honghui Shi, Thomas S. Huang, and Lei Zhang. Higherhrnet: Scale-aware representation learning for bottom-up human pose estimation. *IEEE Conf. Comput. Vis. Pattern Recog.*, 2020. 3

[5] Statefarm Distracted Driver Detection Dataset. https://www.kaggle.com/c/state-farm-distracted-driver-detection/data. 2016. 1

[6] Nachiket Deo and Mohan M. Trivedi. Looking at the driver/rider in autonomous vehicles to predict take-over readiness. *IEEE Transactions on Intelligent Vehicles*, 2019. 1

[7] Tiancheng Hu, Sumit Jha, and Carlos Bus. Robust driver head pose estimation in naturalistic conditions from point-cloud data. *IEEE Intelligent Vehicles Symposium*, 2020. 2

[8] SAE International. Taxonomy and definitions for terms related to driving automation systems for on-road motor vehicles. 2018. 1

[9] Yonghong Xie; Qiang Ji. A new efficient ellipse detection method. *IEEE Object recognition supported by user interaction for service robots*, 2002. 2

[10] Ross Girshick Ali Farhadi Joseph Redmon, Santosh Divvala. You only look once: Unified, real-time object detection. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 4

[11] Ikromjanov Kobiljon Komil Ugli, Ali Hussain, Beom Su Kim, Satyabrata Aich, and Hee-Cheol Kim. A transfer learning approach for identification of distracted driving. In *2021*

23rd International Conference on Advanced Communication Technology (ICACT), pages 1–4, 2021. 2

[12] Martin Kostinger, Paul Wohlhart, Peter M. Roth, and Horst Bischof. Annotated facial landmarks in the wild: A large-scale, real-world database for facial landmark localization. *International Conference on Computer Vision Workshops*, 2011. 3

[13] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. *Adv. Neural Inform. Process. Syst.*, 2012. 2, 3

[14] T. Hoang Ngan Le, Yutong Zheng, Chenchen Zhu, Khoa Luu, and Marios Savvides. Multiple scale faster-rcnn approach to driver's cell-phone usage and hands on steering wheel detection. *IEEE Conf. Comput. Vis. Pattern Recog. Worksh.*, 2016. 2, 3

[15] Maitree Leekha, Mononito Goswami, Rajiv Ratn Shah, Yifang Yin, and Roger Zimmermann. Are you paying attention? detecting distracted driving in real-time. In *2019 IEEE Fifth International Conference on Multimedia Big Data (BigMM)*, pages 171–180, 2019. 2

[16] Manuel Martin, Stephan Stuehmer, Michael Voit, and Rainer Stiefelhagen. Real time driver body pose estimation for novel assistance systems. *IEEE International Conference on Intelligent Transportation Systems*, 2017. 3

[17] Erin Murphy-Chutorian, Anup Doshi, and Mohan M. Trivedi. Head pose estimation for driver assistance systems: A robust algorithm and experimental evaluation. *IEEE Intelligent Intelligent Transportation Systems Conference*, 2007. 2

[18] Mohan M. Trivedi Ofer Achler. Vehicle wheel detector using 2d filter banks. *IEEE Intelligent Vehicles Symposium University of Parma*, 2014. 2

[19] Binbin Qin, Jiangbo Qian, Yu Xin, Baisong Liu, and Yihong Dong. Distracted driver detection based on a cnn with decreasing filter size. *IEEE Transactions on Intelligent Transportation Systems*, pages 1–12, 2021. 2

[20] Akshay Rangesh and Mohan M. Trivedi. Handynet: A one-stop solution to detect, segment, localize analyze driver hands. *IEEE Conference on Computer Vision and Pattern Recognition - 3D HUMANS Workshop*, 2018. 3

[21] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *Int. Conf. Learn. Represent.*, 2015. 2, 3

[22] Ashish Tawari, Sujitha Martin, and Mohan M. Trivedi. Continuous head movement estimator for driver assistance: Issues, algorithms, and on-road evaluations. *IEEE Transactions On Intelligent Transportation Systems*, 15(2), 2014. 2

[23] Tsun-Yi Yang, Yi-Ting Chen, Yen-Yu Lin, and Yung-Yu Chuang. Fsa-net: Learning fine-grained structure aggregation for head pose estimation from a single image. *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019. 2