Latent Body-Pose guided DenseNet for Recognizing Driver's Fine-grained Secondary Activities

Ardhendu Behera and Alexander H Keidel
Department of Computer Science, Edge Hill University, Ormskirk, Lancashire, L39 4QP, UK
{beheraa,keidela}@edgehill.ac.uk

Abstract

Over the past two decades, there has been an increasing research in developing self-driving vehicles, with many industries pushing the bounds alongside academia. Automatic recognition of in-vehicle activities plays a key role in developing such vehicles. In this work, we propose a novel human-pose driven approach for video-based monitoring of driver's state/activity and is inspired by the recent success of deep Convolutional Neural Network (CNN) in visual recognition tasks. The approach infers the driver's state/activity from a single frame and thus, could operate in real-time. We also bring together ideas from recent works on human pose detection and transfer learning for visual recognition. The adapted DenseNet integrates these ideas under one framework, where one stream is focused on the latent body pose and the other stream is on appearance information. The proposed method is extensively evaluated on two challenging datasets consisting various secondary non-driving activities. Our experimental results demonstrate that the driver activity recognition performance improves significantly when the latent body-pose is integrated into the existing deep networks.

1. Introduction

For centuries, self-driving vehicles have intrigued human interest. In the late 1400s, Leonardo da Vinci sketched a hypothetical self-propelled cart and then in 1930s, mechanical autopilot for aeroplanes is emerged. Now, Autonomous Vehicles (AVs) are expected to play a key role in the modern transportation systems since they offer additional safety, increased productivity, greater accessibility and many societal and environmental benefits.

Visual sensor is an integral part of the AVs and is partly due to the significant progress in CAN-bus and imaging technology. There is also immense development in computing power (e.g. multicore, GPU powered computation). These advancement paired with the latest deep learning make it a reality for an autonomous system to collect information and extract relevant knowledge from the environment. Most of these sensing technologies are often focused on surrounding environmental perception and minimal work has focused on the human driver perspectives. It is suggested that mindful attention should be paid to driver behavior [11] since driver error is overwhelmingly to blame for the vast majority of accidents [35, 1]. The driver's role could be replaced by the AVs but the vehicle also requires to deliver performance identical to that of a driver if it is to be trusted [21]. Thus, the AVs should not only focus on the surrounding traffic conditions but also on understanding, modeling and predicting human agents as pointed out in [28].

In highly or fully automated driving, it is found that besides traditional secondary activities (e.g. listening to music, talking to passengers), writing text messages, eating and drinking, browsing the Internet, and calling are most wanted in-vehicle activities [30]. Therefore, there is a need of automatic monitoring of such activities so that the AVs should know the driver's state and readiness for a Take-Over Requests (TOR) [21], defined by the National Highway Traffic Safety Administration (NHTSA). Moreover, automatic recognition of in-vehicle activities would play a prominent role in eventual realization of the *cognitive car* [16] and self-learning AVs [4] concepts. These concepts are aimed to automatically learn from in-vehicle activities to provide a better experience to the occupants and optimize the vehicle's performance.

Human activities/behavior recognition and description in videos and still images is a classic computer vision problem with a long list of literature over the years covering various approaches [17, 29, 14]. However, little progress has been made to adapt these approaches into automobile environment. This could be partly due to the challenge to develop a standard language that can sufficiently and succinctly describe human actions. Nevertheless, there has been some recent progress in machine vision with application to driver behavior/activity monitoring [20, 32, 40]. Most of these approaches are focused on safe driving (e.g. drowsiness,

978-1-5386-9294-3/18/$31.00 ©2018 IEEE

fatigue, lane change intentions, etc). However, very little progress has been made for automatic recognition of in-vehicle secondary activities identified in [30] and will be prominent in highly or fully-automated driving.

In this paper, we propose a novel method for recognizing these activities so that the vehicle would be able to know the driver's current state, which is important for the vehicle to take appropriate decision such as driver's readiness for a TOR. The proposed approach benefits from the potential of *transfer learning* (TL) by using state-of-the-art deep CNN models like DenseNet [19] and Convolutional Pose Machines (CPM) [3, 41]. This work includes the following novel contributions:

1. We present a novel way to combine state-of-the-art latent body-pose with the latest DenseNet [19] model.
2. We demonstrate the effectiveness of this latent body-pose for effective recognition of the fine-grained secondary non-driving activities.
3. We explore the benefits of TL and evaluate the performance on two challenging datasets. We also examine the impact of body-pose on cross-dataset performance.

1.1. Related work

Recognising and describing visual content in images/videos is a fundamental problem in artificial intelligence. Human actions recognition is an important part of this problem. It is still a difficult problem despite a significant progress has been made in recent years. The difficulty is often due to the strong variations of people, their poses and scenes both in motion and appearance. The other common factors include subtle differences in fine-grained actions such as interacting/manipulating small objects as in driver's secondary activities.

Video is a stack of still frames and there is a variety of work in the field of action recognition from static images. The intrinsic property of the existing approaches is to learn mapping functions that link image content to action labels. Majority of these approaches use holistic cues such as hand-object interactions [42, 8, 15], articulated pose [31, 26] and part-based/structured models [43, 7]. These cues often explore the spatial configuration of body joints and hand-object interactions that usually require body parts and object detectors. These approaches are challenged by the recent advances in CNN-based deep models [12, 27, 13]. Most of these deep models learn spatial filters that maximize the recognition accuracy in an end-to-end manner on very large and highly diverse datasets [5, 31]. This allows these models to achieve the highest accuracy. It is unclear about the suitability of these approaches for fine-grained recognition tasks.

Human actions are inherently structured patterns of body pose. Recently, several approaches focus to leverage the pose to guide CNNs [12, 10, 6]. Most of these approaches use the body joints either to pool features [6] or to define an attention mechanism [12, 10]. Therefore, they require the detection of the body joints.

Latest deep CNN models [19, 39, 38] have achieved significant improvement in image recognition tasks using large-scale datasets [34, 24]. Such models consist of multiple layers to learn representation of images with multiple levels of abstraction through a hierarchical learning process [23]. It learns from very general (e.g. edges, color blobs) to task-specific features as we move from first-layer to the last-layer [44, 25]. Thus, these models are recently explored for transfer learning in particular, the transfer of pre-trained network parameters to problems with limited training data [9, 33, 44] and has shown great success.

The traditional vision-based driver monitoring approaches are mainly focused on cues involving upper-body parts (e.g. face, eye, hand and head) and their movements [20, 32, 40]. These approaches are focused on automatic detection of safe/unsafe driving behaviors (e.g. drowsiness, fatigue, lane change intentions, etc.) using hand-crafted features (e.g. LBP, HOG, Haar-like) combined with classical machine learning algorithms such as SVM and AdaBoost. Recently, there is some progress in using CNN models in driver's activities monitoring [2, 18, 36]. However, the adaptation of the state-of-art CNN models driven by the human pose is yet to be explored in the automobile domain. In this paper, we aim to address this.

In this paper, we propose a novel approach that is inspired by the success of the latest CNNs for image recognition and pose-guided action recognition. We adapt the state-of-the-art DenseNet [19] with a simple yet surprisingly powerful modification that benefits from the latent poses for recognizing fine-grained driver's activities. In particular, we explore the benefit of the TL by re-using the latent human poses represented by a set of filters capturing the spatial configuration of body joints. These latent poses are learnt via Part Affinity Fields [3] using MS COCO dataset [24]. We also apply TL by fine-tuning the DenseNet [19] model on the target secondary driving activities dataset.

2. Proposed approach

We aim to recognize driver's fine-grained activities from video frames by adapting the latest DenseNet. Recognizing these activities from still images is a challenging problem and is mainly due to the absence of temporal information. The overall approach is shown in Fig. 1. The aim is to minimize resources (e.g. GPUs) and the required computational time to train/validate the target model while still achieving competitive performance on this fine-grained activity recognition task. Moreover, in real-world applications involving robotics and autonomous vehicles, there is a limitation on power, processing time and computational re-

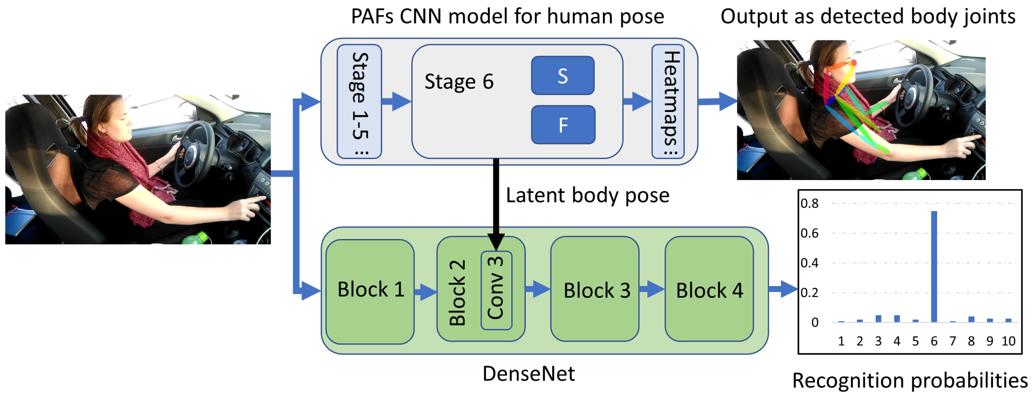


Figure 1: Overview of the adapted DenseNet [19]: An example image (*adjusting radio* activity) from the dataset in [2] is fed. Latent body pose is extracted from the stage 6 using Part Affinity Fields (PAFs) CNN model (pure *transfer learning i.e.* off-the-shelf pre-trained model trained on MS COCO dataset [24]). The latent pose consists of **S**, the confidence maps of body part locations and **F**, which is a 2D vector field encoding the associations between body parts. The latent pose is integrated in the Conv3 layer in Block2 of the DenseNet and trained on the target dataset for fine-grained activity recognition.

sources. Therefore, we infer activity using still frames and is computationally inexpensive. We also adapt the latest DenseNet model in which the number of parameters is significantly less than other state-of-the-art models (reported in [19]). This fits well with the automobile domain involving real-time monitoring for immediate intervention and/or support.

2.1. Adapting DenseNet to include latent pose

A CNN consists of multiple processing layers. When an image I_0 is fed into the model, at each layer l, a non-linear transformation $T_l(.)$ is applied to generate output I_l. $T_l(.)$ can be a single or composite function of operations (e.g. convolution, pooling, dropout, batch normalization (BN) or rectified linear units (ReLU)). Then the output I_l is fed into the next $(l+1)^{th}$ layer which gives rise to the following output $I_{l+1} = T_{l+1}(I_l)$. The procedure continues until the final L^{th} layer to produce the output I_L.

In order to improve the information flow between layers, Dense Convolutional Network (DenseNet) [19] introduced direct connections from any layer to all successive layers. As a result, the layer l receives the image/feature transformations of all preceding layers *i.e.* $I_0, I_1, \cdots I_{l-1}$ as input:

$$I_l = T_l([I_0, I_1, \cdots, I_{l-1}]) \tag{1}$$

where $[I_0, I_1, \cdots I_{l-1}]$ represents the concatenation of the image/feature transformations produced in layers $0, 1, \cdots l-1$.

Inspired by this architecture, we aim to recognize fine-grained driver's activity by integrating human pose in eqn 1. Pose is the configuration of body parts in a given image and is proved to be an important cue in discriminating human activities [12, 10, 6]. Recently, Cao *et al.* [3] proposed a two-branch, six-stages CNN architecture for real-time detection of 2D pose of multiple people in images. The method jointly learns body parts detection and parts association using a non-parametric representation called Part Affinity Fields (PAFs) that encodes both position and orientation of human limbs. At each stage, the first branch predicts a set of 2D confidence maps **S** of body part locations and the second branch anticipates a set of 2D vector fields **F** of part affinities encoding the degree of associations between parts. The goal is to use this model to extract latent poses without fine-tuning/re-training on the target dataset (*i.e.* pure transfer learning). We use the output from the stage 6 in which the latent pose consisting **S** and **F** are represented with 128 filters each. For example, the dimension of both **S** and **F** after stage 6 is $28 \times 28 \times 128$ for an input image size of 224×224. We integrate this with the DenseNet's layer l by concatenating with the input features:

$$I_l = T_l([I_0, I_1, \cdots, I_{l-1}, \mathbf{F}, \mathbf{S}]) \tag{2}$$

Once the latent pose is integrated in the layer l, it will go through all successive layers $(l+1, \cdots, L)$ due to the nature of the direct connections between layers in DenseNet.

In order to maximize the impact of the latent pose, we need to find the appropriate layer l in the DenseNet. It is mentioned earlier that CNN model learns feature from general to specific along the network [44, 25]. In Deep Adaptation Networks (DAN) [25], it is suggested that features representing the middle layers are less transferable, therefore these layers are learned via fine-tuning. Motivated by this, we integrate the body pose in the middle layer (Conv3) of the DenseNet. We also experimented with other layers and found that the network performs better when integrated in the Conv3 layer.

2.2. Training the adapted DenseNet model

We apply the TL approach to train the adapted DenseNet. This is motivated by the findings in [44], which suggests that CNNs perform better when initialized with transferred weights in comparison to random weights while training on a new dataset. Thus, all layers in the DenseNet are initialized with pre-trained ImageNet's [34] weights except the softmax layer (random initialization) due to the different number of classes. The ImageNet consists of 1.2M natural images with 1K categories. The latent pose consisting **S** and **F** is extracted using the pre-trained PAFs CNN model [3] trained on MS COCO dataset [24]. All the layers are frozen (*i.e.* weights do not change during training) and is used as a feed-forward network to extract latent poses.

We use the default image size of 224 × 224 and the data augmentation of height and width shift of up to 20% and a 50% chance for horizontal flip. We train the model with the batch size of 64 using a Linux (Ubuntu) machine fitted with 24GB GPU (NVIDIA Quadro P6000) card. We use stochastic gradient descent (SGD) optimizer to minimize the categorical cross entropy $E_i = -\sum_c y_{i,c} log(p_{i,c})$, where p are the predictions, y are the actual labels, i denote the training images and c represent the fine-grained activities categories. We set an initial learning rate of 0.01, reduced by a factor of 0.9 after every epoch and trained the model for 30 epochs.

3. Experimentation and discussion

We validate our model on two challenging datasets [2, 37] consisting 10 classes: 1) safe driving, 2) texting - right, 3) talking on the phone - right, 4) texting - left, 5) talking on the phone - left, 6) operating the radio, 7) drinking, 8) reaching behind, 9) hair and makeup, and 10) talking to passenger. The dataset in [2] consists of 12,977 training and 4,331 testing images. Whereas, the State Farm [37] dataset has a total of 22,424 training and 79,726 testing images. State Farm's class labels of the training images are available but not for the test images. We manually labeled all the test images.

In our evaluation, we use the accuracy (ACC) and log loss $logLoss = -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{10} y_{i,c} log(p_{i,c})$, where $p_{i,c}$ is the prediction probability of class c, given i^{th} test image and $y_{i,c}$ is actual probability (0 or 1). N is the total number of test images. Accuracy gives the percentage of correct predictions and assigns equal cost to false positives and false negatives. Whereas, log loss quantifies classifier's accuracy by penalizing the confident false classifications. For a perfect classifier, the log loss will be zero.

In our experiments, we evaluate our adapted DenseNet and compare the performance with the original DenseNet [19]. We evaluate on each dataset, as well as the cross-dataset *i.e.* train on one dataset and test on other. This results in four possible combinations on two datasets D_1 [2] and D_2 [37]: 1) Train on train data in D_1 and test on the rest *i.e.* test data in D_1, both train and test data in D_2 (Table 1). 2) Train on train data in D_2 and test on the rest (test data in D_2, both train and test data in D_1) (Table 2).

Method (Trained on train data D_1 [2])	ACC	Loss
DenseNet [19] (Test data D_1)	87.6	0.419
*AlexNet [22] (Test data D_1)	93.7	-
*Inception V3 [39] (Test data D_1)	**95.2**	-
DenseNet [19] (Train data D_2)	47.5	1.939
DenseNet [19] (Test data D_2)	47.8	2.007
Proposed Approach		
DenseNet + Latent pose (Test data D_1)	94.2	**0.233**
DenseNet + Latent pose (Train data D_2)	**55.5**	**1.513**
DenseNet + Latent pose (Test data D_2)	**55.5**	**1.724**

Table 1: Performance when trained using train data in [2] (D_1). Tested on test data in [2] (D_1), as well as cross dataset evaluation *i.e.* tested on both train and test data in [37] (D_2). Accuracy is in percentage. * Results are reported in [2].

3.1. Performance comparison and discussion

The performance of the proposed adapted DenseNet [19] (DenseNet + latent pose) trained using training data in D_1 [2] is shown in Table 1. It is clearly evident that the performance increases ($\backsim$ 7% on D_1 and $\backsim$ 8% on D_2). It is observed that when the model is trained on one dataset and tested on another dataset, the performance is significantly lower in comparison to testing on the same dataset (Table 1, 2). This could be due to the dataset D_1 [2] was collected from seven different countries in four different cars with several variations in driving conditions. Whereas, D_2 [37] consists of images captured using one car in a controlled environment. The other notable difference between the two datasets is that the test data in D_1 consists of images from drivers who are part of the train data as well. Whereas, in D_2, the test data consists of images from completely unseen drivers. Nevertheless, the proposed adapted DenseNet performs better in all cases and is mainly due to the influence of the body pose. This signifies, the body pose plays a key role in discriminating fine-grained activities.

We compare our performance to the state-of-the-art approaches used on these two datasets. On D_1, Abouelnaga *et al.* [2] use AlexNet [22] and Inception V3 [39] and reported the respective performance of 93.65% and 95.17%. We achieve 94.2% using our adapted DenseNets. It is worth noting that we use the lightest DenseNet *i.e.* DenseNet121, which has $\backsim$ 7M parameters. Whereas, in AlexNet [22] and Inception V3 [39], the respective parameters are $\backsim$ 62M and $\backsim$ 22M (Table 3). Therefore, the proposed model is more suitable for mobile and embedded applications such as robotics and autonomous vehicles.

Method (Trained on train data D_2 [37])	ACC	Loss
DenseNet [19] (Test data D_2)	86.1	0.652
†AlexNet [22] (Train data D_2)	72.6	-
†VGG-16 (Train data D_2)	82.5	-
†ResNet-152 (Train data D_2)	85.0	-
DenseNet (Train data D_1)	27.3	4.422
DenseNet (Test data D_1)	27.3	4.496
Proposed Approach		
DenseNet + Latent pose (Test data D_2)	**87.5**	0.843
DenseNet + Latent pose (Train data D_1)	**33.3**	**4.208**
DenseNet + Latent pose (Test data D_1)	**32.3**	**4.266**

Table 2: Performance when trained using train data in [37] (D_2). Tested on test data in [37] (D_2), as well as cross dataset evaluation *i.e*. tested on both train and test data in [2] (D_1). Accuracy is in percentage. † Results are reported in [18] and they have used the train data in D_2 to create their own train (80%) and validation (20%) subset.

CNN models	Number of parameters
AlexNet [22]	62,378,344
Inception V3 [39]	21,823,274
DenseNet201 [19]	18,341,194
DenseNet169 [19]	12,659,530
DenseNet121 [19]	7,047,754

Table 3: Number of parameters involved in various CNN models for recognizing 10 different fine-grained activities.

On D_2, Singh [36] has used VGG19 model on a subset of training images. The reported performance is 10.9% when trained from the scratch and 21.1% when trained using transfer learning. Similarly, Hssayeni *et al.* [18] used 80% of the train data to train AlexNet, VGG16 and ResNet-152. They use the rest 20% of train data to validate the model. They reported the best accuracy of 85% by ResNet-152. Our approach gives 87.5%.

In this experiments, we also evaluated the performance across the datasets and would be helpful for advancing research in domain adaptation problem.

4. Conclusions

We have presented a novel way to integrate human pose into the state-of-the DenseNet model. The adapted DenseNet integrates ideas from human pose estimation and transfer learning to solve fine-grained human activity recognition problems. We have shown that the performance significantly improves with inclusion of the pose and explained its significance. We believe this help advance in fine-grained activity monitoring focusing on AVs and robotics.

Acknowledgments: The research is supported by the Edge Hill University-funded Research Investment Fund (RIF). We are grateful to Taylor Smith in State Farm for providing information about their dataset. The GPU used in this research is generously donated by the NVIDIA.

References

[1] Department of transportation national highway traffic safety administration, traffic safety facts. https://crashstats.nhtsa.dot.gov/Api/Public/ViewPublication/811059, 2008.

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

[3] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *IEEE CVPR*, volume 1, 2017.

[4] O. Carsten. *From Driver Models to Modelling the Driver: What Do We Really Need to Know About the Driver?*, pages 105–120. Springer London, London, 2007.

[5] Y.-W. Chao, Z. Wang, Y. He, J. Wang, and J. Deng. Hico: A benchmark for recognizing human-object interactions in images. In *IEEE ICCV*, pages 1017–1025, 2015.

[6] G. Chéron, I. Laptev, and C. Schmid. P-cnn: Pose-based cnn features for action recognition. In *IEEE ICCV*, pages 3218–3226, 2015.

[7] V. Delaitre, I. Laptev, and J. Sivic. Recognizing human actions in still images: a study of bag-of-features and part-based representations. In *BMVC*, 2010.

[8] C. Desai, D. Ramanan, and C. Fowlkes. Discriminative models for static human-object interactions. In *IEEE CVPR workshops*, pages 9–16, 2010.

[9] J. Donahue et al. Decaf: A deep convolutional activation feature for generic visual recognition. In *ICML*, volume 32, pages 647–655, 2014.

[10] W. Du, Y. Wang, and Y. Qiao. Rpan: An end-to-end recurrent pose-attention network for action recognition in videos. In *IEEE CVPR*, pages 3725–3734, 2017.

[11] D. L. Fisher, M. Lohrenz, D. Moore, E. D. Nadler, and J. K. Pollard. Humans and intelligent vehicles: The hope, the help, and the harm. *IEEE Trans. on Intel. Vehicles*, 1(1):56–67, March 2016.

[12] R. Girdhar and D. Ramanan. Attentional pooling for action recognition. In *Advances in NIPS*, pages 33–44, 2017.

[13] G. Gkioxari, R. Girshick, and J. Malik. Contextual action recognition with r*cnn. In *ICCV*, pages 1080–1088, 2015.

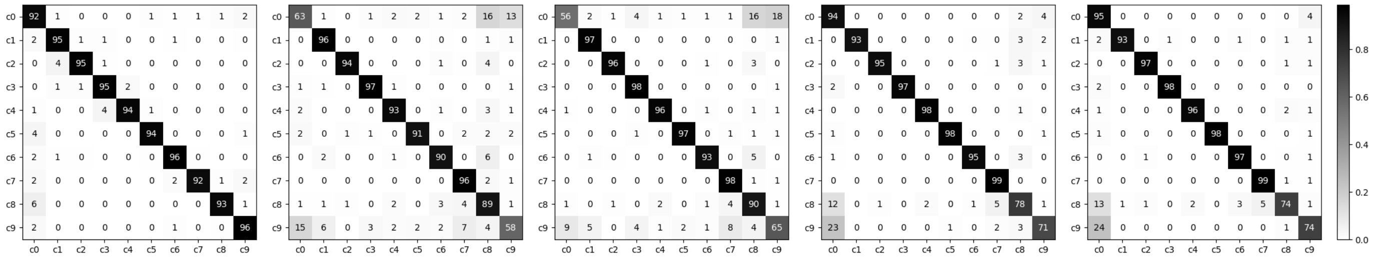
[14] G. Guo and A. Lai. A survey on still image based human action recognition. *Patt. Recog.*, 47(10):3343 – 3361, 2014.

[15] A. Gupta, A. Kembhavi, and L. S. Davis. Observing human-object interactions: Using spatial and functional compatibility for recognition. *IEEE TPAMI*, 31(10):1775–1789, 2009.

[16] A. Heide and K. Henning. The cognitive car: A roadmap for research issues in the automotive sector. *Annual Reviews in Control*, 30(2):197 – 203, 2006.

[17] S. Herath, M. Harandi, and F. Porikli. Going deeper into action recognition: A survey. *IVC*, 60:4 – 21, 2017.

[18] M. Hssayeni, S. Saxena, R. Ptucha, and A. Savakis. Distracted driver detection: Deep learning vs handcrafted features. *Electronic Imaging*, 2017(10), 2017.



(a) DenseNet + pose (D_1) (b) DenseNet (D_2) (c) DenseNet + pose (D_2) (d) DenseNet ($D_2^{'}$) (e) DenseNet + pose ($D_2^{'}$)

Figure 2: Confusion matrices (y-axis: actual and x-axis: predicted) for recognizing fine-grained activities). D_1 represents the model is trained and tested on dataset in [2]. D_2 represents it is trained and tested on dataset in [37]. $D_2^{'}$ demonstrates the model is trained on *test data* and tested on *train data* in [37].

[19] G. Huang, Z. Liu, K. Q. Weinberger, and L. van der Maaten. Densely connected convolutional networks. In *IEEE CVPR*, volume 1, pages 4700–4708, 2017.

[20] S. Kaplan, M. A. Guvensan, A. G. Yavuz, and Y. Karalurt. Driver behavior analysis for safe driving: A survey. *IEEE Trans. on Intel. Transp. Syst.*, 16(6):3017–3032, Dec 2015.

[21] H. J. Kim and J. H. Yang. Takeover requests in simulated partially autonomous vehicles considering human factors. *IEEE Trans. on Human-Machine Syst.*, 47(5):735–740, Oct 2017.

[22] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in NIPS*, 2012.

[23] Y. LeCun, Y. Bengio, and G. E. Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

[24] T.-Y. Lin et al. Microsoft coco: Common objects in context. In *ECCV*, pages 740–755, 2014.

[25] M. Long, Y. Cao, J. Wang, and M. I. Jordan. Learning transferable features with deep adaptation networks. *arXiv preprint arXiv:1502.02791*, 2015.

[26] S. Maji, L. Bourdev, and J. Malik. Action recognition from a distributed representation of pose and appearance. In *IEEE CVPR*, pages 3177–3184, 2011.

[27] A. Mallya and S. Lazebnik. Learning models for actions and person-object interactions with transfer to question answering. In *ECCV*, pages 414–428, 2016.

[28] E. Ohn-Bar and M. M. Trivedi. Looking at humans in the age of self-driving and highly automated vehicles. *IEEE Trans. on Intel. Vehicles*, 1(1):90–104, March 2016.

[29] L. Onofri, P. Soda, M. Pechenizkiy, and G. Iannello. A survey on using domain and contextual knowledge for human activity recognition in video streams. *Expert Syststems with Applications*, 63:97–111, 2016.

[30] B. Pfleging, M. Rang, and N. Broy. Investigating user needs for non-driving-related activities during automated driving. In *Proc. of the 15th Int'l Conf. on mobile and ubiquitous multimedia*, pages 91–99. ACM, 2016.

[31] L. Pishchulin, M. Andriluka, and B. Schiele. Fine-grained activity recognition with holistic and pose based features. In *German Conf. on Pattern Recog.*, pages 678–689, 2014.

[32] B. Ranft and C. Stiller. The role of machine vision for intelligent vehicles. *IEEE Transactions on Intelligent Vehicles*, 1(1):8–19, 2016.

[33] A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson. Cnn features off-the-shelf: An astounding baseline for recognition. In *IEEE CVPRW*, pages 512–519, 2014.

[34] O. Russakovsky et al. ImageNet Large Scale Visual Recognition Challenge. *IJCV*, 115(3):211–252, 2015.

[35] B. D. Seppelt et al. Glass half-full: On-road glance metrics differentiate crashes from near-crashes in the 100-car data. *Accident Analysis & Prevention*, 107:48 – 62, 2017.

[36] D. Singh. Using convolutional neural networks to perform classification on state farm insurance driver images. Technical report, Stanford University, 650 Serra Mall, CA, 2016.

[37] State Farm Corporate. State farm distracted driver detection, April 2016, `www.kaggle.com/c/state-farm-distracted-driver-detection`.

[38] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *AAAI*, pages 4278–4284, 2017.

[39] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. In *IEEE CVPR*, pages 2818–2826, 2016.

[40] M. M. Trivedi, T. Gandhi, and J. McCall. Looking-in and looking-out of a vehicle: Computer-vision-based enhanced vehicle safety. *IEEE Trans. on Intel. Transp. Syst.*, 8(1):108–120, March 2007.

[41] S.-E. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh. Convolutional pose machines. In *IEEE CVPR*, pages 4724–4732, 2016.

[42] B. Yao and L. Fei-Fei. Modeling mutual context of object and human pose in human-object interaction activities. In *IEEE CVPR*, pages 17–24, 2010.

[43] B. Yao, X. Jiang, A. Khosla, A. L. Lin, L. Guibas, and L. Fei-Fei. Human action recognition by learning bases of action attributes and parts. In *IEEE ICCV*, pages 1331–1338, 2011.

[44] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson. How transferable are features in deep neural networks? In *Advances in NIPS*, pages 3320–3328, 2014.