Deep Passenger State Monitoring using Viewpoint Warping

Ian Tu[1(✉)], Abhir Bhalerao[1], Nathan Griffiths[1], Mauricio Muñoz[2], Thomas Popham[2], and Alex Mouzakitis[2]

[1] Department of Computer Science, University of Warwick, Coventry, UK
{i.tu,abhir.bhalerao,nathan.griffiths}@warwick.ac.uk
[2] Jaguar Land Rover, Engineering Centre, Coventry, UK
{amunozd1,tpopham,amouzak1}@jaguarlandrover.com

Abstract. The advent of autonomous and semi-autonomous vehicles has meant passengers now play a more significant role in the safety and comfort of vehicle journeys. In this paper, we propose a deep learning method to monitor and classify passenger state with camera data. The training of a convolutional neural network is supplemented by data captured from vehicle occupants in different seats and from different viewpoints. Existing driver data or data from one vehicle is augmented by viewpoint warping using planar homography, which does not require knowledge of the source camera parameters, and overcomes the need to re-train the model with large amounts of additional data. To analyse the performance of our approach, data is collected on occupants in two different vehicles, from different viewpoints inside the vehicle. We show that the inclusion of the additional training data and augmentation by homography increases the average passenger state classification rate by 11.1%. We conclude by proposing how occupant state may be used holistically for activity recognition and intention prediction for intelligent vehicle features.

Keywords: Passenger State Monitoring · Camera Homography · Convolutional Neural Networks · Classification · Deep Learning

1 Introduction

With the advent of autonomous and semi-autonomous vehicles, and smart vehicles systems, all the occupants inside a vehicle have become relevant to safety and comfort. In recent years, the focus has been mainly on monitoring the driver, for example whether they are alert [16], or paying attention to the road [20]. In self-driving vehicles, the type of behaviour and actions being monitored are not limited to safe driving and it is important for the vehicle to sense if the driver is in a state to regain control if required, so called hand-over from autonomous driving to driver control. Knowing the state and behaviour of the occupants, including the driver, is also useful for optimising the in-vehicle experience, which involves monitoring the state of all occupants. Many car manufacturers, such as Jaguar Land Rover, are developing and building new intelligent vehicle systems

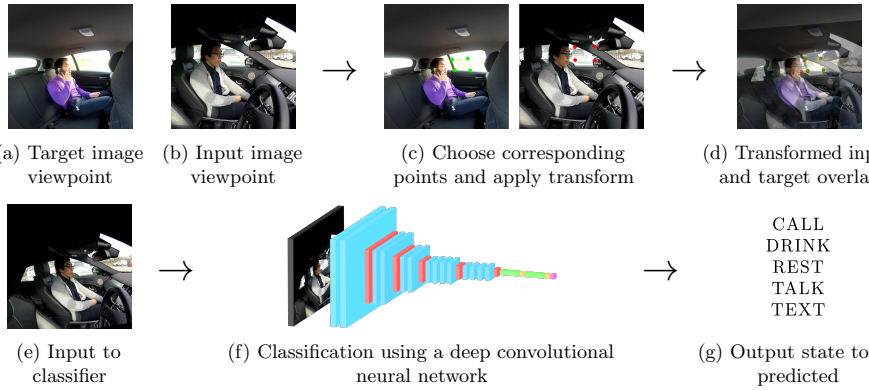
(e.g. ADAS [9]) for semi-autonomous and connected vehicles, and being able to monitor and predict occupant state is a vital parameter in designing, optimising and adapting a car's intelligent vehicle systems so as to maximise the safety and comfort of a journey [6].

A way to observe and analyse the actions and behaviours of vehicle occupants is to use inward facing cameras, which have the advantage that they are relatively cheap and general purpose sensors. Computer vision-based methods are proposed where image features are detected as proxies of driver state. The signals captured from imagery have been shown to be robust in identifying driver fatigue and distraction, for example see [10] [2].

Deep learning using Convolutions Neural Networks (CNNs) classification methods have been demonstrated to work effectively for a variety of visual object classification problems, provided there is sufficient training data available, an appropriate deep architecture can be realised that generalises well to unseen data, e.g. [13]. CNNs have been shown to be more effective and have less overfitting when the size of the training data is increased [3] [17]. One way to increase the amount of training data is to augment the data, data augmentation is the process of transforming the data without altering the data's labels - a common practice in visual-based problems is to apply geometric transformations such as rotations, scaling, flips, etc. [13].

Recent related works involving driver behaviour include a paper by Yan [23], using the Southeast University Driving-posture Dataset (SEU dataset) by [25] designed a CNN model to identify 6 driver actions: calling, eating, braking, wheel use, phone use, and smoking. They used 2 inputs to the CNN, a primary input and a secondary input; the primary input was a bounding box around the whole driver in the image, and the secondary input was a set of skin regions. Their method was successful in determining the correct action, even when two actions were very similar, achieving a mean average precision of 97.8%. A CNN trained on 4 different driver postures, driving, answering phone call, eating and smoking, with an overall accuracy of 99.8% was used in [21]. For driver gaze detection, in the main step determines which of the 9 different gaze zones the driver was looking at. This method achieves an average of 95% accuracy [4]. Another method, [22] uses a CNN model to predict driver fatigue and distraction from the locating the eye, ears and mouth, and achieves overall accuracy of 95.6% in classifying the six states: eyes open/closed, mouth normal/eating, and ear normal/on phone.

Though actions are usually associated with movement and so video is required, it transpires that many simple actions can be identified from only still imagery. Deep learning utilising CNNs for action recognition has proved effective at image classification and object detection which rely heavily on image features [18] [11] [7]. An action can be determined from contextual cues, such as a person's pose or the presence of an object. For example, Gkioxari's [7] action recognition method exploits this by marking bounding boxes around the subject and bounding boxes around relevant contextual cues and then using these as inputs to a CNN, in a manner similar to Yan's driver monitoring method [23].



(a) Target image viewpoint (b) Input image viewpoint (c) Choose corresponding points and apply transform (d) Transformed input and target overlay

(e) Input to classifier (f) Classification using a deep convolutional neural network (g) Output state to be predicted

Fig. 1: The passenger state classification pipeline: (a),(b) A target viewpoint is chosen and all remaining image data will be aligned to that viewpoint using planar homography, (c); (d) Comparison of target and test viewpoints; (e)-(g) Deep state classification using a trained CNN.

Because passenger state monitoring and action detection is still in its infancy and focused on the driver, there is limited data available depicting passenger state, and much of this data relevant for driver behaviour, and action classification pertinent to vehicle safety. In this paper, we: (1) estimate projective transformations (planar homography) [8] to compute approximate viewpoint corrections to re-use driver monitoring data for passenger state classification; (2) widen the set of states detected to those of interest to semi-autonomous and autonomous vehicles; (3) augment the data set to generalise for small camera motions, including homography, and; (4) train and analyse the performance of a CNN for passenger state classification. We demonstrate that the approach has a number of benefits: it enables the large amounts of existing driver state monitoring data to be re-used for occupant monitoring in general so explicit changes in camera positions can be made without the need to recapture passenger data. It also gives flexibility to the transfer the learning model between vehicles; by augmenting the data set with randomised projective transformations, the learnt model also becomes more robust to small view-point changes and generalises better.

2 Method

The proposed method has two stages: image alignment and classification using a convolutional neural network. The image alignment stage chooses a single viewpoint to which all the other images are mapped and this mapping is calculated by marking corresponding points from two example images. The output of this stage is a view-transformed dataset where all images are approximately from the same viewpoint. The second stage consist of re-training a partially-trained CNN model, augmenting the training samples with small viewpoint variations, regressed to labelled output state, see Figure 1.

2.1 Image Alignment Using Homography

A homography is a projective transformation from one plane to another and can be defined as the algebraic linear mapping $h : \mathbb{R}^2 \mapsto \mathbb{R}^2$ is a homography if and only if there exist a non-singular 3×3 matrix H such that for any point in $\mathbb{R}^2$, represented by a homogeneous coordinate x, $h(\mathrm{x}) = \mathrm{Hx}$ [8]. We can express the mapping of a 2D point (x, y) in homogeneous coordinates as a vector $\mathrm{x} = (x, y, 1)^T$, and likewise a target 2D point (u, v) as $\mathrm{u} = (u, v, 1)^T$, through a homography matrix $\mathbf{h} = (h_{ij})$:

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} \sim \begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \tag{1}$$

This mapping can be solved using the Direct Linear Transform (DLT) algorithm. The homography matrix $\mathbf{h}$ is scale-invariant, meaning multiplying by a non-zero constant will not change the equations needed to be solved, so $\mathbf{h}$ is a homogeneous matrix representing only 8 degrees of freedom, therefore, there are only 8 unknowns to solve for. As a result, 4 pairs of (non-colinear) points are required, with each pair of source and target points providing two equations:

$$\begin{aligned} xh_{11} + yh_{12} + h_{13} - xuh_{31} - yuh_{32} - uh_{33} = 0 \\ yh_{21} + yh_{22} + h_{23} - xvh_{31} - yvh_{32} - vh_{33} = 0 \end{aligned} \tag{2}$$

The homographic matrix $\mathbf{h}$ can be found by solving, $A_i\mathbf{h} = \mathbf{0}$, with SVD, where

$$A_i = \begin{pmatrix} x_i & y_i & 1 & 0 & 0 & 0 & -u_i x_i & -u_i y_i & -u_i \\ 0 & 0 & 0 & x_i & y_i & 1 & -v_i x_i & -v_i y_i & -v_i \end{pmatrix}. \tag{3}$$

More accurate estimates of $\mathbf{h}$ are obtained with more that 4 point pairs, though to obtain a single homogeneous solution all the corresponding point pairs need to be exact. However, in most cases, the point pairs are inexact, so a suitable cost function will need to be minimised to solve for $\mathbf{h}$ [8].

The use of homography in real world applications range from camera calibration to the 3D reconstruction of a scene using images from different camera viewpoints [24] [5]. In most of these cases, the input and target images are from the same scene, so similar feature correspondences such as similar points between images the images can be found, e.g. using SURF descriptors [1] to stitch together a panorama [12]. However, in the context of vehicle occupant monitoring, the images from one dataset could be significantly different from another dataset, so using a feature detection method will not be effective. Therefore, input and target images from the corresponding datasets would require manual labelling. Only a single input and target image needs be chosen, under the assumption that for each dataset the images are from a fixed or similar camera viewpoints, otherwise further homography matrices are needed from the additional images with different camera viewpoints. Using the homography matrix calculated from the input and target images, all the images from corresponding input dataset are transformed to be similar to the target dataset image's viewpoint. The resulting datasets are then be used for training the CNN model, Figure 1a to Figure 1e shows results of the image alignment process on two different datasets.

2.2 Synthesising Viewpoint Changes

For data augmentation, we apply image warping to our training data set by randomising using homography given knowledge of the intrinsic camera matrix of the target viewpoint. The camera projection (without lens distortion) is modelled as by the perfect pin-hole camera geometry such that 3D world coordinate points $\mathbf{X}$ project to the camera plane as $\mathbf{x}$ through the product of the intrinsic and extrinsic camera matrices, K and $(R|T)$,

$$\begin{pmatrix} \mathbf{x} \\ 1 \end{pmatrix} \sim K \begin{pmatrix} R & T \\ \mathbf{0}^T & 1 \end{pmatrix} \begin{pmatrix} \mathbf{X} \\ 1 \end{pmatrix} \tag{4}$$

To synthesise small viewpoint changes around the principal axis of the camera i.e. $T = \mathbf{0}$, we induce random motions of the principal axis and rotations around this axis. Letting the quaternion, Q, represent the spatial rotation of θ about the principal axis, $(a_x, a_y, 1)^T$, the (z-axis) in camera coordinates:

$$Q = (a_x sin(\theta/2), a_y sin(\theta/2), sin(\theta/2), cos(\theta/2)) \tag{5}$$

then, random motions of the axis and random rotations can be created by drawing normal random samples $a_x, a_y \sim \mathcal{N}(0, \sigma_{xy})$ and $\theta \sim \mathcal{N}(0, \sigma_\theta)$. After normalisation, the rotations $Q(\sigma_{xy}, \sigma_\theta)$ are then converted to the matrices R and substituted into Equation 4 to generate the 3×3 random homography matrix, $H = K(R|0)$.

2.3 Passenger State Classification

Convolutional neural networks (CNNs) are machine learning models which use multiple or deep neural network layer, combining a number of operational layers. They can be trained to learn regressions of pixel values from images and used as supervised classifiers to learn object classes [14]. Below, we detail the CNN network architecture and training regime.

Network architecture. The CNN model architecture is based on the commonly used VGG19 architecture [19] as it has been shown generalise well on a wide range of datasets. The input to the network was a 224×224 RGB image, with the output being one the 5 states: calling on phone, drinking, resting, talking and texting. The network architecture used is outlined in Figure 2. To prevent overfitting, dropout with $p = 0.5$ was applied to the fully connected layers.

Transfer learning. The passenger state classification CNN was pre-trained on the ILSVRC 2012 dataset (ImageNet) is retrained to work on our occupant state datasets. In order to incorporate the model for our own use, the fully connected layer weights were discarded in the pre-trained model, and randomly initialised weights were used for these layers. The number of epochs (iteration over entire dataset) ranged from $10 - 100$. A smaller learning rate was used for the new model's weights thus effectively fine-tuning the results. The learning rate was set initially to 1e$-$3 and decreased accordingly after $5 - 10$ epochs, weight decay was set to 1e$-$6, and was trained in batches of 16 using stochastic gradient descent (SGD) [15] with momentum 0.9.

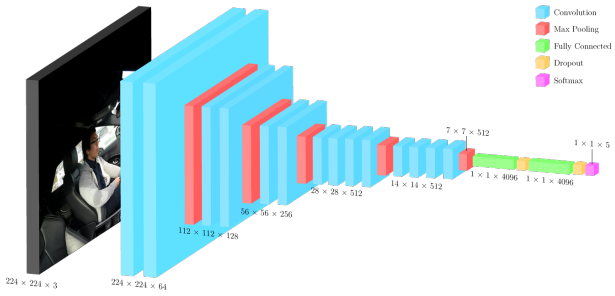


Fig. 2: Modified VGG19 network architecture.

3 Experiments and Results

3.1 Dataset

We conducted experiments to create two different datasets, one dataset captured in a Land Rover SUV and another BMW hatchback. Participants were asked to conduct typical in-vehicle actions whilst being filmed; these actions included mobile phone use, eating, drinking, sleeping, and talking. At the end of data collection, the video data was converted into 1920×1080 images, and each image was labelled with a state. States where there was not enough data were excluded from the final dataset. The states that are included are the following:

- **Call**: The occupant has their mobile phone up to their face for a phone call.
- **Drink**: The occupant is drinking or in the motion of drinking.
- **Rest**: The occupant is not engaging in any notable activity, this includes sleeping.
- **Talk**: The occupant is actively engaging in a conversation with another occupant.
- **Text**: The occupant is looking at their phone and actively using their phone.

The amount of data and the distribution of data is as following:

- **Land Rover SUV** dataset. This dataset contains 7 people enacting 5 states. There are 36151 images, from 3 different viewpoints, the approximate distribution of images to class is 30/5/20/15/30 respectively for call/drink/rest/talk/text.
- **BMW Hatchback** dataset. This dataset contains 8 people enacting 5 states. There are 30340 images, from 2 different viewpoints, the approximate distribution of images to class is 12/8/30/20/30 respectively for call/drink/rest/talk/text.

Figure 3 shows example images from these datasets and their viewpoints.

3.2 Evaluation

For each dataset, for each viewpoint, we split the images into the following subsets: 80% of images for training and 20% of images for testing. The training data is further split into 80% for training and 20% for validation. The data is split according to individual, so for example, training using the Land Rover data will mean 4 individuals are used for training, 1 for validation, and 3 for testing. There

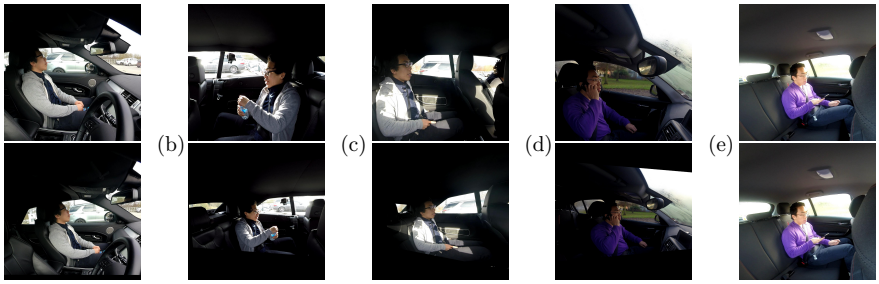


Fig. 3: The upper row shows the original images with the 4 different viewpoints from the camera situated at: (a) Front seat right in the Land Rover (rest state); (b) Back seat left in the Land Rover (drink state); (c) Back seat right in the Land Rover (talk state); (d) Front seat right in the BMW (call state); (e) Back seat right in the BMW (text state). The bottom row shows the transformed images, (e) is the reference viewpoint.

is no overlap in individuals for the training, validation, and testing splits. There is no overlap in individuals for splits between vehicles. The images are randomly picked from each individual and each class for training, validation, and testing.

We combine the two datasets into one using the proposed homography alignment. We choose an image from the training set of the BMW back right seat camera viewpoint as the target image, then all other images not from that viewpoint are mapped to that specified target image using the proposed homography method. The RGB images are then cropped into a $1:1$ ratio, by equal clipping of the sides, and resized to a resolution of 224×224. The proposed CNN models are then trained, validated and tested with a different combination of viewpoints and datasets, and using different weight initialisations. No data augmentation is used apart from the horizontal flip required to map the back seat left viewpoint to back seat right viewpoint, and also the proposed homography alignment.

The performance is compared amongst 9 different models, each is repeated 3 times with a different split of individuals. The average accuracy of the repeated models will be used as a guide of performance. The models are the following:

A) A model trained only using **BMW**. **No homography alignment**. **ImageNet** weight initialisation. Individual train/validation/test split is 4/1/3, the number of images in the split is 4000/1000/1250 respectively.
B) A model trained only using data from the **Land Rover**. Horizontal flip. **No homography alignment**. **ImageNet** weight initialisation. Individual train/validation/test split is 3/1/3, the number of images in the split is 6000/1500/1875.
C) A model trained only using data from the **Land Rover**. Horizontal flip. **No homography alignment**. **Model A** weight initialisation. Individual train/validation/test split is 3/1/3, the number of images in the split is 6000/1500/1875.
D) A model trained only using data from the **BMW**. **Homography alignment**. **ImageNet** weight initialisation. Individual train/validation/test split is 4/1/3, the number of images in the split is 4000/1000/1250.
E) A model trained only using data from the **Land Rover**. Horizontal flip. **Homography alignment**. **ImageNet** weight initialisation. Individual train/validation/test split is 3/1/3, the number of images in the split is 6000/1500/1875.

F) A model trained only using data from the **Land Rover**. Horizontal flip. **Homography alignment**. **Model D** weight initialisation. Individual train/validation/test split is 3/1/3, the number of images in the split is 6000/1500/1875.

G) A model trained using data from the **BMW** and **Land Rover**, but only using **front seat** viewpoint images. **Homography alignment**. **ImageNet** weight initialisation. Individual train/validation/test split is 10/2/3, the number of images in the split is 4000/1000/1250.

H) A model trained using data from the **BMW** and **Land Rover**, but only using back seat viewpoint images. Horizontal flip. **Homography alignment**. **ImageNet** weight initialisation. Individual train/validation/test split is 10/2/3, the number of images in the split is 6000/1500/1875.

I) A model trained using data from the **BMW** and **Land Rover**, but only using **back seat** viewpoint images. Horizontal flip. **Homography alignment**. **Model G** weight initialisation. Individual train/validation/test split is 10/2/3, the number of images in the split is 6000/1500/1875.

3.3 Results

Table 1 shows the accuracy results in confusion matrix form for every model. For the models with no homography alignment applied, the performance is poor; models A (Table 1a), B (Table 1b) and C (Table 1c) score under 60% for overall average accuracy. The models particularly struggle to classify the rest and talk states, often wrongly predicting between the two. This a persistent problem for all of the models. Figure 4c shows that the models find it difficult to distinguish between open and closed mouth states for certain individuals because there are many similarities between these two states, and given a small image it may not be possible to classify accurately.

The significance of using more data can be seen in the results for Model C; when the weights are used from Model A to help train Model C, there is an improvement of 5.2%, increasing the accuracy from 52.0% to 57.2%. The results for Model C in Table 1c, shows a large increase in accuracy for the talking class, up from 59% to 75%, albeit at the cost of more classes being misclassified talking - the texting class notably suffers the most from this. All other classes, except for the texting state, show a minor improvement from using Model A's weights.

Models D (Table 1d), E (Table 1e) and F (Table 1f) are A, B and C's respective counterparts and are trained using aligned data benefit from applying the proposed homography alignment process. These models show a substantial improvement, ranging from an increase of 5.3% to 18.1% in overall accuracy. The call state accuracy for these models is significantly improved. Although, the call labelled images are mistakenly misclassified as the talk state, see for example Figure 4a where the presence of a phone is not easily discernible and the mouth is a much more prominent feature - the converse also applies.

Model F uses Model D's weights as initialisation of its weights and helps increase the overall average accuracy by a significant 11.1% from 64.2% in Model E, where it just uses ImageNet weights, to 75.3%. The classes all show improvement except for drinking. The drinking class images exhibit higher misclassification as the resting class. An example of this is shown in Figure 4b as the passenger

Table 1: Confusion matrices for evaluation models: (a)-(c) trained on non-aligned data; (d)-(f) trained with viewpoint-aligned data; (g)-(i) front seat and back seat state classification (also aligned data). See main text for details.

(a) Model A (**58.8%**)

Call	Drink	Rest	Talk	Text
0.49	0.32	0.00	0.19	0.00
0.01	0.73	0.03	0.10	0.13
0.00	0.06	0.32	0.54	0.08
0.02	0.07	0.32	0.51	0.09
0.00	0.06	0.00	0.04	0.89

(b) Model B (**52.0%**)

Call	Drink	Rest	Talk	Text
0.27	0.22	0.17	0.25	0.08
0.00	0.71	0.12	0.04	0.13
0.02	0.10	0.44	0.36	0.07
0.02	0.09	0.22	0.59	0.07
0.00	0.16	0.18	0.07	0.59

(c) Model C (**57.2%**)

Call	Drink	Rest	Talk	Text
0.36	0.15	0.09	0.32	0.08
0.01	0.72	0.10	0.12	0.05
0.02	0.11	0.55	0.28	0.04
0.01	0.09	0.12	0.75	0.03
0.00	0.19	0.16	0.16	0.49

(d) Model D (**64.1%**)

Call	Drink	Rest	Talk	Text
0.77	0.05	0.03	0.10	0.05
0.14	0.81	0.00	0.04	0.01
0.02	0.12	0.46	0.31	0.08
0.09	0.05	0.29	0.30	0.26
0.00	0.10	0.01	0.03	0.86

(e) Model E (**64.2%**)

Call	Drink	Rest	Talk	Text
0.65	0.10	0.19	0.05	0.01
0.01	0.83	0.08	0.05	0.03
0.04	0.04	0.76	0.11	0.05
0.01	0.07	0.49	0.42	0.02
0.00	0.20	0.19	0.05	0.56

(f) Model F (**75.3%**)

Call	Drink	Rest	Talk	Text
0.73	0.01	0.15	0.10	0.01
0.02	0.75	0.14	0.06	0.03
0.02	0.05	0.79	0.08	0.06
0.02	0.02	0.22	0.68	0.06
0.00	0.04	0.10	0.05	0.82

(g) Model G (**65.5%**)

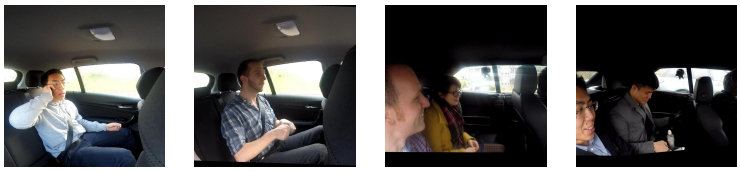
Call	Drink	Rest	Talk	Text
0.59	0.32	0.00	0.06	0.03
0.03	0.85	0.08	0.02	0.03
0.02	0.05	0.55	0.29	0.10
0.00	0.01	0.39	0.41	0.19
0.00	0.08	0.02	0.03	0.88

(h) Model H (**68.6%**)

Call	Drink	Rest	Talk	Text
0.71	0.00	0.19	0.09	0.01
0.03	0.66	0.11	0.09	0.11
0.02	0.05	0.70	0.13	0.11
0.02	0.02	0.28	0.61	0.07
0.00	0.08	0.11	0.07	0.74

(i) Model I (**75.3%**)

Call	Drink	Rest	Talk	Text
0.89	0.01	0.07	0.03	0.01
0.07	0.73	0.08	0.04	0.08
0.02	0.03	0.71	0.13	0.11
0.03	0.02	0.22	0.65	0.08
0.00	0.08	0.11	0.04	0.77



(a) Talk/Call (b) Rest/Drink (c) Rest/Talk (d) Talk/Text

Fig. 4: Misclassified images. Predicted label/true label.

performs an unexpected bottle opening action, the model incorrectly classifies it as the closest looking state, resting. This is symptom of not having enough data, as noted in Section 3.2, the drinking state has the fewest images compared to other classes. In contrast to Model E, Model F shows a major increase in the text state, 56% to 82%, but this is still sometimes misclassified as the talk state. An example of this is shown in Figure 4d, in some cases the images do not provide sufficient information, such as the phone not being fully visible in the camera field of view.

Model G (Table 1g) is trained on the aligned front seat data from both vehicles, Model H (Table 1h) and Model I (Table 1i) are trained in aligned, back seat views from both vehicles. The transfer of weights from the front seat model results in an increase for all states for the back seat model with the overall average accuracy rising by 6.7% from 68.6% to 75.3%. Even though Model G originally shows difficulty in discerning the call class, when used to help train the back seat model, it notably improves the call state classification accuracy from 71% to 89%.

4 Conclusions

In this paper, we propose a passenger state detection method that uses a convolutional neural network in combination with viewpoint warping using planar homography. This enables data which usually cannot be included at the training stage to be effectively used for re-training, and data re-purposed from driver monitoring to occupant state classification. The viewpoint normalisation and augmentation also allows the trained model to be re-trained with additional data to work between vehicle types. To evaluate the robustness of the proposed method, data was collected in two different vehicles at three different viewpoints, and used to demonstrate that viewpoint is a significant factor influencing accuracy. The results show that there is a benefit to using data from other vehicles and other viewpoints through transfer learning. Furthermore, we show that it is possible to usefully apply data from the driver monitoring to passenger state monitoring.

Being able to accurately classify passenger state, albeit to a limited number of distinct classes, opens up possibilities to build driving state monitoring systems for passenger-to-passenger and driver-to-passenger interactions. The current results are limited to well-lit vehicle cabins, and data was only captured with a relatively small number of people and two vehicle models. To assess the flexibility and robustness of the approach, future work will focus on further passenger states, data will be collected in a larger range of vehicle types, and under more demanding lighting environments, such as during night-time journeys.

Acknowledgement This work was supported by Jaguar Land Rover and the UK-EPSRC grant EP/N012380/1 as part of the jointly funded Towards Autonomy: Smart and Connected Control (TASCC) Programme. We wish to thank all who volunteered to take part in the data collection.

References

1. Bay, H., Ess, A., Tuytelaars, T., Van Gool, L.: Speeded-up robust features (surf). Computer vision and image understanding 110(3), 346–359 (2008)
2. Bergasa, L.M., Nuevo, J., Sotelo, M.A., Barea, R., Lopez, M.E.: Real-time system for monitoring driver vigilance. IEEE Trans. on ITS 7(1), 63–77 (2006)
3. Chatfield, K., Simonyan, K., Vedaldi, A., Zisserman, A.: Return of the devil in the details: Delving deep into convolutional nets. In: BMVC (2014)

4. Choi, I.H., Tran, T.B.H., Kim, Y.G.: Real-time categorization of driver's gaze zone and head pose using the convolutional neural network. In: Proc. of HCI Korea. pp. 417–422. Hanbit Media, Inc. (2016)
5. Chuan, Z., Da Long, T., Feng, Z., Li, D.Z.: A planar homography estimation method for camera calibration. In: Proc. of IEEE Comp. Int. in Robotics and Automation. vol. 1, pp. 424–429. IEEE (2003)
6. Daza, I.G., Bergasa, L.M., Bronte, S., Yebes, J.J., Almazán, J., Arroyo, R.: Fusion of optimized indicators from Advanced Driver Assistance Systems (ADAS) for driver drowsiness detection. Sensors 14(1), 1106–1131 (2014)
7. Gkioxari, G., Girshick, R., Malik, J.: Contextual action recognition with r* cnn. In: Proc. of the IEEE ICCV. pp. 1080–1088 (2015)
8. Hartley, R., Zisserman, A.: Multiple view geometry in computer vision. Cambridge university press (2003)
9. Jaguar Land Rover: New ADAS technologies for 2017 Range Rover Sport. http://media.landrover.com/node/10699 (2016), (Accessed on June 18, 2017)
10. Ji, Q., Zhu, Z., Lan, P.: Real-time nonintrusive monitoring and prediction of driver fatigue. IEEE Trans. on Vech. Tech. 53(4), 1052–1068 (2004)
11. Ji, S., Xu, W., Yang, M., Yu, K.: 3d convolutional neural networks for human action recognition. IEEE Trans. on PAMI 35(1), 221–231 (2013)
12. Juan, L., Oubong, G.: Surf applied in panorama image stitching. In: Proc. of Image Proc. Theory Tools and Apps. pp. 495–499. IEEE (2010)
13. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: NIPS. pp. 1097–1105 (2012)
14. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proc. of the IEEE 86(11), 2278–2324 (1998)
15. LeCun, Y.A., Bottou, L., Orr, G.B., Müller, K.R.: Efficient backprop. In: Neural networks: Tricks of the trade, pp. 9–48. Springer (2012)
16. Mbouna, R.O., Kong, S.G., Chun, M.G.: Visual analysis of eye state and head pose for driver alertness monitoring. IEEE Trans. on ITS 14(3), 1462–1469 (2013)
17. McLaughlin, N., Del Rincon, J.M., Miller, P.: Data-augmentation for reducing dataset bias in person re-identification. In: AVSS. pp. 1–6. IEEE (2015)
18. Simonyan, K., Zisserman, A.: Two-stream convolutional networks for action recognition in videos. In: NIPS. pp. 568–576 (2014)
19. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
20. Vicente, F., Huang, Z., Xiong, X., De la Torre, F., Zhang, W., Levi, D.: Driver gaze tracking and eyes off the road detection system. IEEE Trans. on ITS 16(4), 2014–2027 (2015)
21. Yan, C., Coenen, F., Zhang, B.: Driving posture recognition by convolutional neural networks. IET Computer Vision 10(2), 103–114 (2016)
22. Yan, C., Jiang, H., Zhang, B., Coenen, F.: Recognizing driver inattention by convolutional neural networks. In: CISP. pp. 680–685. IEEE (2015)
23. Yan, S., Teng, Y., Smith, J.S., Zhang, B.: Driver behavior recognition based on deep convolutional neural networks. In: ICNC-FSKD. pp. 636–641. IEEE (2016)
24. Zhang, Z.: A flexible new technique for camera calibration. IEEE Trans. on PAMI 22(11), 1330–1334 (2000)
25. Zhao, C., Zhang, B., He, J., Lian, J.: Recognition of driving postures by contourlet transform and random forests. IET Int. Trans. Sys. 6(2), 161–168 (2012)