Dual Viewpoint Passenger State Classification Using 3D CNNs

Ian Tu[1], Abhir Bhalerao[1], Nathan Griffiths[1], Mauricio Muñoz Delgado[2],
Alasdair Thomason[2], Thomas Popham[3], Alex Mouzakitis[2]

Abstract—**The rise of intelligent vehicle systems will lead to more human-machine interactions and so there is a need to create a bridge between the system and the actions and behaviours of the people inside the vehicle. In this paper, we propose a dual camera setup to monitor the actions and behaviour of vehicle passengers and a deep learning architecture which can utilise video data to classify a range of actions. The method incorporates two different views as input to a 3D convolutional network and uses transfer learning from other action recognition data. The performance of this method is evaluated using an in-vehicle dataset, which contains video recordings of people performing a range of common in-vehicle actions. We show that the combination of transfer learning and using dual viewpoints in a 3D action recognition network offers an increase in classification accuracy of action classes with distinct poses, e.g. mobile phone use and sleeping, whilst it does not apply as well for classifying those actions with small movements, such as talking and eating.**

I. Introduction

The fast pace of advances in technology has provided vehicles with an ever increasing range of intelligent features designed to aid drivers and passengers and help them have safe and comfortable vehicle journeys. At the current rate of development, driving itself will become an optional experience in the near future and the vehicle will take you to your desired destination without manual intervention. With fully-autonomous vehicles everyone inside the vehicle will thus be relegated to being a passenger, and so there is an increasing need to find ways to tailor the experience to the needs of a passenger, rather than to the needs of a driver only.

Currently, features such as adaptive cruise control and ADAS technologies such as self-parking improves a driver's performance, but with the shift to autonomy, there will eventually be just as many convenience features applicable to everyone inside the vehicle and so accurate occupant state monitoring is essential. Monitoring actions, gestures and behaviours can improve the experience from the moment an occupant enters the vehicle. For example, if the vehicle recognises a person, it could automatically adjust the seat position, and during a journey if a person is asleep, the vehicle might change the ride settings to give maximal comfort. The observation of actions and behaviours needs to be seamless and non-invasive: not require the need to wear sensors on the person. Video cameras combined with machine learning algorithms are ideal for this purpose: they are relatively cheap and unobtrusive, and computer vision methods can be effectively applied to recognise human poses and gestures.

Passenger state monitoring aims to recognise actions and behaviours of vehicle occupants to help personalise and enhance the vehicle experience. Occupant state monitoring, focused on passengers, is in its infancy with current literature using single shot images rather than video [1]. In this paper we propose: (1) a 3D convolutional neural network (CNN) model which analyses video data to classify common actions inside a vehicle; (2) utilise two different viewpoints from two different cameras inside the vehicle and evaluate performance of the 3D CNN model; (3) analyse whether data from the domain of action recognition can be applied to vehicle occupant state monitoring by using transfer learning.

II. Related Work

Vision-based action recognition is a popular field of research as it has a number of applications, ranging from surveillance and behaviour analysis [2] to facilitating human-computer interactions [3]. For vehicle applications, cameras have been used to improve the safety of the occupants: external cameras can be used to detect and avoid pedestrians and other road users [4], whilst in-vehicle cameras have been used to detect driver state to enable a vehicle to alert the driver when they are becoming inattentive, distracted or drowsy [5].

A. Action Recognition

In the past, motion-based descriptors such as histograms of optical flow (HOF) and motion boundary histograms (MBH) have been employed to recognise actions [6]. Recently, however, from their success in still-image recognition tasks [7] deep learning techniques have been increasingly applied to action recognition.

Simmoyan and Zisserman [8] presented a novel CNN approach to action recognition, this method gives an increase in performance over motion-based descriptor methods scoring 88.0% overall accuracy compared to 85.9% of [6] in UCF-101 [9], a widely used action dataset. They developed a two-stream model which averages the prediction gained from a

[1] Ian Tu, Abhir Bhalerao, and Nathan Griffiths are with the Department of Computer Science, University of Warwick, Coventry, UK. Email: {i.tu,abhir.bhalerao,nathan.griffiths}@warwick.ac.uk

[2] Mauricio Muñoz Delgado, Alasdair Thomason and Alex Mouzakitis are with Jaguar Land Rover, Engineering Centre, Coventry, UK. Email: {amunozd1,athomaso,amouzak1} @jaguarlandrover.com

[3] Thomas Popham is with the School of Engineering, University of Warwick, Coventry, UK. Email: t.popham@warwick.ac.uk

single RGB image frame and also a collection of optical flow frames, with each individual prediction made through a pre-trained ImageNet 2D CNN [7]. This method enables the network to gain temporal information from the optical flow in addition to spatial information from RGB image channels.

Alternatively, others have attempted a more direct approach using a CNN which extracts directly the temporal and spatial information from the raw video data. In order to analyse 3D data, they apply 3D convolutions, hence building a 3D CNN. The work of Tran et al. [10] has popularised the 3D architecture, called the C3D model, it achieves 82.3% in UCF-101. Further improvements to this model by others, such as [11], have shown that increasing the temporal length of inputs and incorporating optical flow features improves the performance to 92.7% in UCF-101.

Further accuracy improvements have been achieved by pre-training on a large dataset beforehand, shown in [12], they also created a new model I3D. I3D utilises RGB and optical flow and is shown to have state-of-the-art results when pre-trained on a database which consists of 400 action classes, (Kinetics) [13]. RGB-I3D, the version of I3D which only uses RGB frames as input, achieves 95.6% overall accuracy in UCF-101, while the two-stream I3D achieves 98.0% accuracy with Kinetics pre-trained weights.

Recently, the 3D ResNet architecture has been shown to achieve better results than C3D and have comparable results to RGB-I3D if Kinetics is used for pre-training, with 3D ResNeXt attaining 94.5% accuracy in UCF-101 [14]. This further reinforces the observation that deep 3D architectures are effective for video action recognition, though with the caveat that it helps to pre-train on a sufficiently large database such as Kinetics. Additionally, since 3D ResNeXt's input is four times smaller than RGB-I3D, the model is more efficient than the RGB-I3D whilst still achieving similar performance.

B. Occupant State Monitoring

Occupant state monitoring is the detection of the actions and behaviours of all the people that are inside a vehicle. Using this information, the vehicle can be adapted and personalised to suit each occupant's individual taste, to improve the in-vehicle experience. Currently, most of these monitoring methods have been developed to solely aid the driver, detecting whether a driver is fatigued or distracted in order to prevent accidents. Following their success on general action recognition problems, occupant state monitoring frameworks have also started to employ deep learning methods.

For example, in [5] a CNN model is used to detect phone activity and whether a driver's hands are on the wheel by locating the eye, ears and mouth regions. Next, these regions are used as inputs to a CNN to classify one of the six states: eyes open/closed, mouth normal/eating, and ear normal/on phone. Their experimental results show that they achieve an overall accuracy of 95.6%.

Yan [15] created a CNN model which can classify driver state directly from image data and [16] uses the whole image and the hand, face and skin regions as input. The hand, face and skin regions are separately detected also using a CNN beforehand. The method achieves a remarkable accuracy of 99.8% in recognising pose.

In a recent paper by Tu et al. [1], passengers are incorporated into the occupant state monitoring model. They use a CNN model to directly predict the states of passengers from an RGB image. The input data is aligned to correspond with images of the training data in order to improve performance. The passenger states predicted are: calling on a mobile phone, drinking, resting, talking and mobile phone use in hand. The overall accuracy of the method is 75.3%.

With these methods, we see that CNNs are capable of being used as feature detectors for specific areas such as the face and hands, and they are also shown to be useful when analysing an entire image. Furthermore, they have been shown to be applicable to the vehicle occupant state monitoring domain, although presently their use is limited to single frame image data.

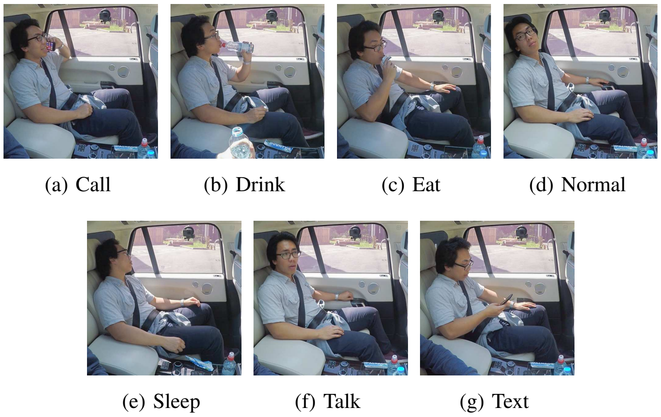
C. Driver/Passenger State Monitoring Datasets

For driver monitoring there are two significant datasets:

- **Southeast University Driving-posture Dataset (SEU dataset) [17]** - This contains 6 driver actions: calling, eating, braking, wheel use, phone use, and smoking. There are 20 participants: 10 male and 10 female.
- **Driver Distraction Dataset [16]** - This recently published dataset features 10 driver distraction actions, which include talking, mobile use, reaching for items and drinking. There are 31 participants (22 male and 9 female), filmed in 4 different vehicles, containing a total of $17,308$ frames.

For action recognition evaluation, there are several commonly used datasets for training and benchmarking machine learning models: HMDB-51 [18] and UCF-101 [9] have been available since the early years of action recognition research and are regularly used as benchmarking methods, even though the consensus is that they do not contain enough data to train deep CNNs. Datasets such as Activity-Net [19] were created to build more accurate action recognition models, although Activity-Net still has insufficient data to create robust CNN models. The Kinetics [20] dataset was acquired for this purpose and contains an abundant quantity of video data for training. The profiles of these datasets are summarised below:

- **UCF-101** - Contains 101 action classes. There are $13,320$ clips and the average duration of each video is approximately 7 seconds.
- **HMDB-51** - Contains 51 action classes. There are $6,766$ videos and the average duration of each video is approximately 3 seconds.
- **Activity-Net** - Contains 200 action classes. There are approximately 137 videos for each class, $28,108$ total action instances and there is about 849 hours of video.
- **Kinetics** - Contains 400 action classes. There are more than 400 videos for each class and the total number of frames is greater than $300,000$.



(a) Call (b) Drink (c) Eat (d) Normal

(e) Sleep (f) Talk (g) Text

Fig. 1. Passenger state image examples.

The dataset used in this paper is new:

- **Warwick Passenger State Monitoring Dataset** - The dataset contains 13 individuals and there are 7 labelled actions. The dataset has over 250,000 frames. More details are outlined III-B Data Collection.

III. Experimental Setup

A. Cameras

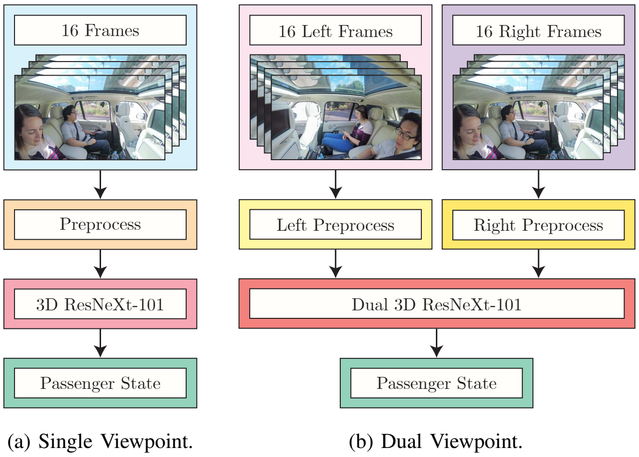
Two identical GoPro Hero 5 cameras were used in the experiments, each held in place on the rear passenger windows using suction cup accessories. The cameras were placed at the top right corner of the window to obtain a wide field of view and to capture the occupant in the seat furthest from the camera. The video resolution of the RGB cameras is 4K at a size of 3840×2160, the footage was filmed at a 30 FPS, using ISO 400 with the default automatic exposure settings.

B. Data Collection

The videos were filmed in a full-sized Sport Utility Vehicle (SUV) while stationary. Subjects were asked to act out a range of actions, including talking to each other, mobile phone use, eating and drinking. The SUV dual viewpoint passenger action dataset contains the following states:

1) **Call** - The subject acts out the process of picking up a phone call, talking on a phone call and ending the call.
2) **Drink** - The subject obtains their drink container from where it was resting and takes a drink out of it and places it back to its location.
3) **Eat** - The subject obtains a small item of food (e.g. a chocolate bar or sandwich) and eats it, at the end the subject discards any waste into a compartment.
4) **Normal** - The subject is in a neutral state and not performing any of the other actions.
5) **Sleep** - The subject pretends to sleep.
6) **Talk** - The subject is conversing with another passenger.
7) **Text** - The subject is using their phone (excludes the process of dialling/selecting a phone number).

Figure 1 shows image examples of the various states from the dataset. The dataset consists of 13 unique individuals, 9 male and 4 female, with each individual having approximately 15 minutes of video of them enacting out various actions, totalling over 250,000 frames.



(a) Single Viewpoint. (b) Dual Viewpoint.

Fig. 2. The passenger state classification pipelines: (a) The pipeline for a single viewpoint model; (b) The pipeline for a dual viewpoint model.

IV. Method

To detect a passenger action state a set of 16 frames are taken from the video data off a single camera, this set is then propagated through a 3D CNN to classify a single action state, one of the 7 states: calling, drinking, eating, normal, sleeping, talking, or texting. The single viewpoint method is shown in Figure 2a. Similarly for a dual viewpoint model, 16 frames are extracted from each viewpoint to be processed by a dual 3D CNN, this is shown in Figure 2b.

A. Network Architecture

There have been great advances in recent years towards 2D image recognition CNNs whereas 3D networks architectures have been less common and have generally not performed as well as their 2D counterparts. Recently for action recognition, it was shown that 3D CNN architectures can be successfully applied [10] [13] [14]. One of the methods has shown that the ResNet architectures can be applicable to video action recognition as long as they are pre-trained on a large action recognition dataset [14].

These ResNet models perform comparably with other action recognition 2D and 3D architectures, with some variations of the ResNet architecture performing better than the state of the art. The principal idea behind the ResNet architecture is that it provides a shortcut between layers which enabling training of even deeper networks [21]. The variation used here is the ResNeXt-101 architecture [22] which has been shown to have the best accuracies when pre-trained with the Kinetics dataset. ResNeXt introduces cardinality, referring to the number of convolutional groups located in the central ResNet block, and is shown to be more effective than using deeper or wider networks.

As analysing two viewpoints in parallel is already very computationally expensive without the burden of estimating optical flow as well, a 3D architecture using only RGB frame data is preferred in this case. Moreover, 3D ResNet

TABLE I
SINGLE VIEWPOINT 3D RESNEXT-101 ARCHITECTURE

Stage	3D ResNeXt-101
conv1	$7 \times 7 \times 7$, 64, stride 2
conv2	$3 \times 3 \times 3$ max pool, stride 2
	$\begin{bmatrix} 1 \times 1 \times 1, 128 \\ 3 \times 3 \times 3, 128 \\ 1 \times 1 \times 1, 256 \end{bmatrix} \times 3$
conv3	$\begin{bmatrix} 1 \times 1 \times 1, 256 \\ 3 \times 3 \times 3, 256 \\ 1 \times 1 \times 1, 512 \end{bmatrix} \times 4$
conv4	$\begin{bmatrix} 1 \times 1 \times 1, 512 \\ 3 \times 3 \times 3, 512 \\ 1 \times 1 \times 1, 1024 \end{bmatrix} \times 23$
conv5	$\begin{bmatrix} 1 \times 1 \times 1, 1024 \\ 3 \times 3 \times 3, 1024 \\ 1 \times 1 \times 1, 2048 \end{bmatrix} \times 3$
	global average pool, 7-d fc, softmax

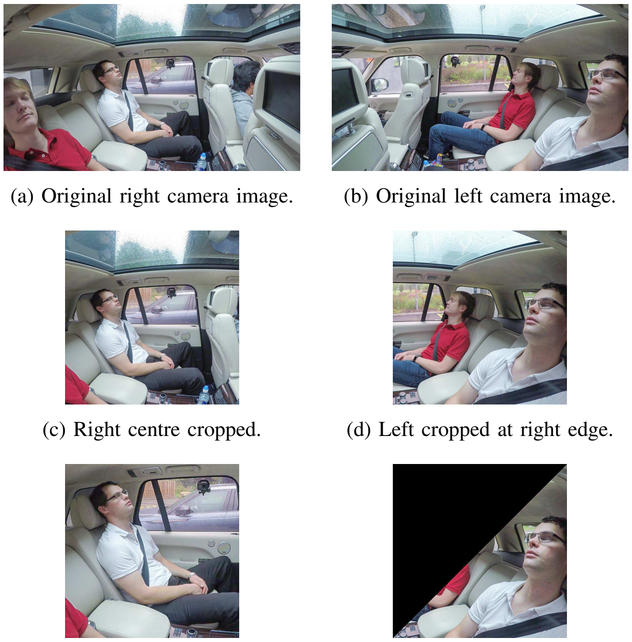
architectures that use only RGB data, when pre-trained on the Kinetics data beforehand, have been shown to have comparable performance to the two-stream 2D RGB-Optical flow architectures [20].

Here, a passenger state classification 3D ResNeXt CNN was pre-trained on the Kinetics dataset and is retrained to work with our vehicle passenger video dataset. To customise the network to fit the 7 classes in the single viewpoint model, the fully pre-trained connected layer weights are discarded and replaced with a new fully connected layer with randomly initialised weights and 7 outputs. The model architecture for a single viewpoint is shown in Table I. Each ResNeXt residual block [22] performs grouped convolutions with 32 groups, and after every convolutional layer there is batch normalization and a ReLU activation layer. There is also downsampling, with a stride of two, that is performed in the first ResNeXt residual block in convolution layers $3, 4$ and 5.

In the dual viewpoint model, for each view a separate 3D ResNeXt CNN is built and trained. Then pre-trained weights are reused, and the fully connected layers are replaced with a fresh fully connected layer with 512 units. The outputs of both these networks are then joined by final fully connected layer with 1024 features and 7 outputs, representing the 7 actions states.

B. Dataset Preprocessing

We take advantage of the fact that the camera positions are fixed and so the video can be pre-processed to make it easier to classify. The right viewpoint video data is cropped and zoomed in a way to focus only on the subject, leaving out background information so that the model can focus mainly on the subjects' movements and not on the surroundings. Figure 3e shows an example of the final input from the right viewpoint. Moreover, for the left viewpoint video data, the other side of the vehicle can be masked out leaving only the subject in view. This is to help prevent the network learning what the other passenger is doing as it may be detrimental in determining a subject's actions. Figure 3f shows an example of the masked left viewpoint.



(a) Original right camera image.

(b) Original left camera image.

(c) Right centre cropped.

(d) Left cropped at right edge.

(e) Right image zoomed.

(f) Left image masked.

Fig. 3. To get an input from the right camera, the original right image (a) is centre cropped, result shown in (c), and then zoomed in to get (e). To get an input from the left camera, the original left image (b) is square cropped from the rightmost edge, result shown in (d), and afterwards a triangular mask is applied on the top left of the image to get (f).

C. Training

To choose training samples the video clips were converted into frames at a frame rate of 30 FPS. Clips ranged from a few seconds to a few minutes, each with a single action. To obtain a 16-frame segment, a random starting point was uniformly chosen in the video clip. Then 16 consecutive frames were selected as a data sample, wrapping around to the start of the segment should less than 16 frames remain.

The procedure to augment the data was similar to that described in [23]. Multi-scale cropping was used, with crops starting at either each corner or the centre of the frame image, the scale being one of $\{100\%, 90\%, 80\%, 70\%, 60\%, 50\%\}$. The mean of the dataset used in training is subtracted from each colour channel in each individual frame. Each colour frame image is cropped and scaled to have an aspect ratio of $1 : 1$ and have an image size of 112×112. The total size of each input image is $3 \times 16 \times 112 \times 112$, which represents channels, frames, width and height respectively.

The number of epochs, iterations over the entire dataset, ranged from $200 - 500$. A small learning rate was use to fine-tune the pre-trained model, the learning rate was initially set as $1\mathrm{e}{-3}$ and if the validation loss does not show change after 10 epochs the learning rate is scaled down by a factor of 10. The training is performed in batches of 32 for the single

TABLE II
MODEL A - RIGHT VIEWPOINT MODEL
F1-SCORE: 50.9%

Call	Drink	Eat	Norm	Sleep	Talk	Text
0.98	0.01	0.00	0.00	0.00	0.00	0.00
0.02	0.98	0.00	0.00	0.00	0.00	0.00
0.37	0.52	0.04	0.05	0.00	0.00	0.02
0.12	0.00	0.00	0.30	0.00	0.41	0.16
0.00	0.00	0.00	0.49	0.24	0.27	0.00
0.15	0.00	0.00	0.21	0.00	0.44	0.20
0.00	0.00	0.03	0.01	0.01	0.00	0.95

TABLE III
MODEL B - LEFT VIEWPOINT MODEL
F1-SCORE: 48.3%

Call	Drink	Eat	Norm	Sleep	Talk	Text
0.71	0.05	0.00	0.25	0.00	0.00	0.00
0.00	0.99	0.01	0.00	0.00	0.00	0.00
0.14	0.63	0.14	0.09	0.00	0.00	0.00
0.04	0.00	0.02	0.89	0.00	0.03	0.02
0.00	0.00	0.00	0.61	0.39	0.00	0.00
0.10	0.00	0.03	0.66	0.00	0.12	0.10
0.09	0.00	0.03	0.36	0.00	0.00	0.51

TABLE IV
MODEL C - DUAL VIEWPOINT MODEL
F1-SCORE: 60.6%

Call	Drink	Eat	Norm	Sleep	Talk	Text
0.97	0.00	0.00	0.03	0.00	0.00	0.00
0.00	1.00	0.00	0.00	0.00	0.00	0.00
0.00	0.89	0.10	0.01	0.00	0.00	0.00
0.01	0.00	0.00	0.54	0.00	0.35	0.10
0.00	0.00	0.00	0.00	1.00	0.00	0.00
0.01	0.00	0.00	0.42	0.00	0.39	0.18
0.00	0.00	0.02	0.01	0.00	0.01	0.96

TABLE V
MODEL D - PRE-TRAINED DUAL VIEWPOINT MODEL
F1-SCORE: 74.7%

Call	Drink	Eat	Norm	Sleep	Talk	Text
0.99	0.00	0.00	0.00	0.00	0.00	0.00
0.00	1.00	0.00	0.00	0.00	0.00	0.00
0.00	0.42	0.56	0.02	0.00	0.00	0.01
0.10	0.00	0.00	0.55	0.00	0.23	0.12
0.00	0.00	0.00	0.25	0.75	0.00	0.00
0.14	0.00	0.00	0.33	0.00	0.36	0.17
0.00	0.01	0.01	0.02	0.00	0.00	0.95

viewpoint models and batches of 16 for the dual viewpoint models. Cross-entropy loss is used, and the network is trained using stochastic gradient descent (SGD) with momentum 0.9. The weight decay is set to $1e{-}3$.

D. Testing

The video clips are split into one second segments and it is verified by hand that they contain one of the seven actions. These shortened video clips can either contain the start of the action, the end of the action or anything in between. The video segments are then converted to frames, with the frame rate at 30 FPS, so each one second segment contains 30 frames. A sliding window method is then adopted to obtain the class label: the first 16 frames are inputted into the network and the output label scores are stored, the next 16 frames are chosen with the first 8 frames being from the previous segment's and the output of this is also stored, and so on. At the end, the output score of all the 16 frame segments are averaged and the maximum of this is the action label given to the clip.

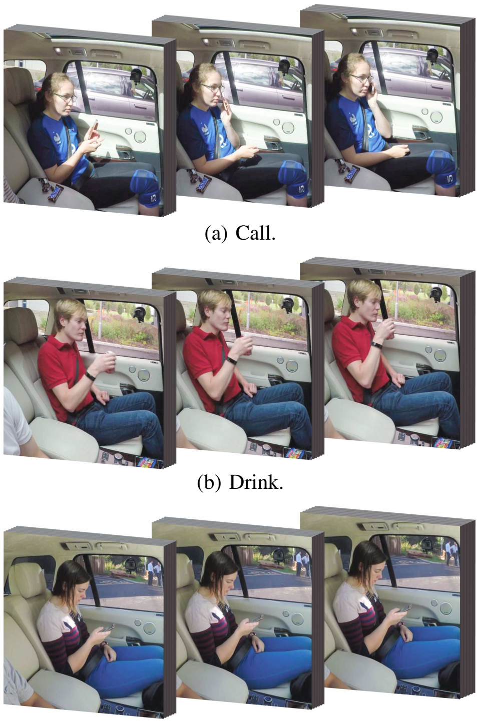
V. RESULTS

Models are trained using various viewpoints and initialised with various weights. For all models, the training, validation and testing uses a unique individual split: there are 13 unique individuals and the split for train/validation/test is $7/2/4$. Tables II to V show the results in confusion matrices form for the following models:

a) **Model A** - This model is trained exclusively on only video from the ***right*** camera. The model is initialised with ***Kinetics*** weights.
b) **Model B** - This model is trained exclusively on only video from the ***left*** camera. The model is initialised with ***Kinetics*** weights.
c) **Model C** - This model is trained on video from ***both*** cameras. The model is initialised with ***Kinetics*** weights.
d) **Model D** - This model is trained on video from ***both*** cameras. The model is initialised with ***Model A's*** and ***Model B's*** weights.

A. Right Viewpoint

Table II shows the results of Model A, the right viewpoint model. On the whole, the model performs poorly, achieving a weighted F1-Score of 50.9%. However, it performs particularly well on the call, drink and text state, with these states reaching accuracy rates of 95% or higher. The actions of these classes are very distinct and so the model has little difficulty in labelling these actions even with a small input resolution from the video stream. Figure 4 shows examples of these correct classification inputs. However, this is not the case with the normal, sleeping and talking states, as these actions score poorly with accuracies between 24% and 44%. The sleeping class is being misclassified between the normal and talk states. A reason why the accuracies of these three actions are relatively low is because the input resolution may be too poor to perceive any detailed changes. For example, the movements during talking represent only small changes in

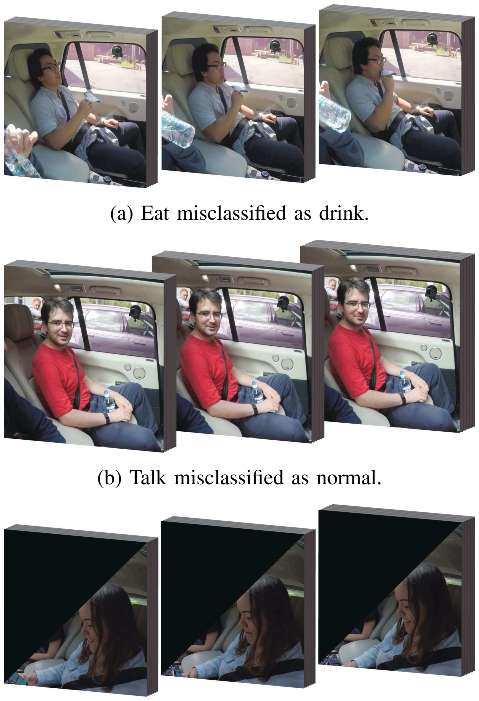


(a) Call.

(b) Drink.

(c) Text.

Fig. 4. Correctly classified input images.



(a) Eat misclassified as drink.

(b) Talk misclassified as normal.

(c) Text misclassified as normal.

Fig. 5. Misclassified input images.

the spatial and temporal dimensions in 112×112 frames. An example of this misclassification is shown in Figure 5b. The eating class also performs poorly on right viewpoint video input, with only 4% accuracy and is mostly misclassified as the call or drink state. A possible explanation for this is that again the resolution is too small to see any object a person is holding, and so the model defaults to the drink case, or the model misclassifies the object as a phone instead. Another reason the model misclassifies the eat state as the call state, is that it may be observing that the raising arm action during eating is similar to the raising arm action of taking a phone call. Figure 5a shows an eat misclassification case where it could be that either the model considers the motion to be similar to drinking, or that the item is more similar to a drink than what it perceives as food, this may also be misclassified because there is a drink in the image as well.

B. Left Viewpoint

Table III shows the results of Model B, the left viewpoint model. Overall, this model also performs just as poorly, achieving a weighted F1-Score of 48.3%. It only performs well in the drink and normal states, having accuracy rates of 99% and 89% respectively. Although for the normal state, the model classifies this as the default when it cannot choose an action correctly it appears to default to the normal state. Compared to the Model A, the right viewpoint model, the eating class again gets misclassified as the drink state or call state, again for similar reasons. A downside of using only the left viewpoint is that it obscures a subject's hands, as most subjects were right-handed the call and text are not seen fully and so this model achieves accuracies of 71% and 51% respectively (compared to the right viewpoint model where the accuracies are almost perfect). Figure 5c shows an example of this obscurement. Unfortunately, even with the view of the subject's face being clearer, since the left viewpoint video view is largely a close-up of the subject's face, the action states which might be thought to improve, such as eating, sleeping and talking, only show a minor improvement. This may be a resolution effect or more likely to do with insufficient training data for those classes.

C. Dual Viewpoint

Table IV shows the results of Model C which takes both the left and right viewpoints as input to a dual viewpoint 3D CNN. Overall, this model achieves a weighted F1-Score of 60.6%, approximately a 10% improvement over single viewpoint models. It performs exceptionally well on the call, drink, sleep and text states achieving accuracy rates of 96% or higher. These actions show very noticeable body movement and body positions and since the model has both viewpoints it does not find difficulty in discerning them. The eat class performs poorly, achieving a very low 14% accuracy rate, with it being largely misclassified as the drink state. Thus, the model is probably classifying eating objects as drinking objects. Furthermore, despite having two viewpoints the normal and talk actions are still being mixed up, which seems to support the hypothesis that a video resolution of 112×112 is insufficient to reliably discern facial movements.

Table V shows the results of Model D which takes both viewpoints and uses the weights of the previous models, Model A and Model B, to initialise the dual viewpoint 3D CNN. Previously, the models were only initialised with

Kinetics weights, in this case the model is initialised with weights from earlier models to improve the convergence to an optimal solution. Model D achieves a weighted F1-Score of 74.7%, a 14.1% improvement compared to initialising from weights which are not tailored to action recognition inside vehicles. This model suffers similar misclassification problems as Model C, the one initialised using Kinetics, although now the eat class has an accuracy rate of 56%. Compared to Model C the sleep class accuracy has fallen from 100% to 75%, with misclassification as the normal class. A possible reason why is that model is giving greater weight to the normal class when the subject is still rather than concentrating on the eyes, although of course since the resolution is small it is often difficult to see the changes in eye state.

VI. Conclusions

We propose a method for recognising common actions of passengers inside the vehicle. It consists of using inward facing cameras on each side of the vehicle. The video data from both cameras is then used as input into a 3D convolution neural network, the output being a single distinct action state. The use of a 3D CNN helps incorporate temporal information of actions as well as the spatial information. Additionally, the use of dual viewpoints aids the model in overcoming occlusions and perceiving more detail from a subject from a different viewpoint. Furthermore, the CNN model was pre-trained on a general action recognition dataset beforehand, demonstrating that transfer learning can be used from help the model gain better recognition rates. For evaluation, data was collected in a stationary vehicle with subjects enacting various common actions and behaviours, which ranged from talking to mobile use. The results show that there is an advantage in using dual viewpoints and applying transfer learning to action video data from a vehicle.

This paper has demonstrated that a multiple camera approach for in-vehicle passenger state recognition is feasible, and could enable camera systems to be successfully incorporated into systems which non-invasively interface with passengers. Future work will address the limitations of the presented approaches, involving collecting and analysing data from moving vehicles. We will also evaluate near IR cameras and operation in low-light and night-time driving. Furthermore, additional subjects will be recruited, and data will be collected in different types of vehicles to evaluate the robustness of 3D CNN classification. A limitation to the accuracy of the system to eating and drinking states might be overcome by incorporating a third channel of information taken from the face regions of the two viewpoints, and this will be investigated.

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] I. Tu, A. Bhalerao, N. Griffiths, M. Delgado, T. Popham, and A. Mouzakitis, "Deep passenger state monitoring using viewpoint warping," in *International Conference on Image Analysis and Processing*. Springer, 2017, pp. 137–148.

[2] J. Shao, K. Kang, C. Change Loy, and X. Wang, "Deeply learned attributes for crowded scene understanding," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 4657–4666.

[3] S. S. Rautaray and A. Agrawal, "Vision based hand gesture recognition for human computer interaction: a survey," *Artificial Intelligence Review*, vol. 43, no. 1, pp. 1–54, 2015.

[4] S. Yi, H. Li, and X. Wang, "Pedestrian behavior understanding and prediction with deep neural networks," in *European Conference on Computer Vision*. Springer, 2016, pp. 263–279.

[5] C. Yan, H. Jiang, B. Zhang, and F. Coenen, "Recognizing driver inattention by convolutional neural networks," in *CISP*. IEEE, 2015, pp. 680–685.

[6] H. Wang and C. Schmid, "Action recognition with improved trajectories," in *Computer Vision (ICCV), 2013 IEEE International Conference on*. IEEE, 2013, pp. 3551–3558.

[7] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[8] ——, "Two-stream convolutional networks for action recognition in videos," in *Advances in neural information processing systems*, 2014, pp. 568–576.

[9] K. Soomro, A. R. Zamir, and M. Shah, "Ucf101: A dataset of 101 human actions classes from videos in the wild," *arXiv preprint arXiv:1212.0402*, 2012.

[10] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3d convolutional networks," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 4489–4497.

[11] G. Varol, I. Laptev, and C. Schmid, "Long-term temporal convolutions for action recognition," *arXiv preprint arXiv:1604.04494*, 2016.

[12] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.

[13] J. Carreira and A. Zisserman, "Quo vadis, action recognition? a new model and the kinetics dataset," *arXiv preprint arXiv:1705.07750*, 2017.

[14] K. Hara, H. Kataoka, and Y. Satoh, "Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet?" *arXiv preprint arXiv:1711.09577*, 2017.

[15] C. Yan, F. Coenen, and B. Zhang, "Driving posture recognition by convolutional neural networks," *IET Computer Vision*, vol. 10, no. 2, pp. 103–114, 2016.

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

[17] C. Zhao, B. Zhang, J. He, and J. Lian, "Recognition of driving postures by contourlet transform and random forests," *IET Int. Trans. Sys.*, vol. 6, no. 2, pp. 161–168, 2012.

[18] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre, "Hmdb: a large video database for human motion recognition," in *Computer Vision (ICCV), 2011 IEEE International Conference on*. IEEE, 2011, pp. 2556–2563.

[19] F. Caba Heilbron, V. Escorcia, B. Ghanem, and J. Carlos Niebles, "Activitynet: A large-scale video benchmark for human activity understanding," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 961–970.

[20] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev *et al.*, "The kinetics human action video dataset," *arXiv preprint arXiv:1705.06950*, 2017.

[21] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[22] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2017, pp. 5987–5995.

[23] L. Wang, Y. Xiong, Z. Wang, and Y. Qiao, "Towards good practices for very deep two-stream convnets," *arXiv preprint arXiv:1507.02159*, 2015.