


ORIGINAL RESEARCH

DIPNet: Driver intention prediction for a safe takeover transition in autonomous vehicles

 Mahdi Bonyani¹ | Mina Rahmanian² | Simindokht Jahangard³ | Mahdi Rezaei⁴ 
¹Department of Computer Engineering, University of Tabriz, Tabriz, Iran

²Department of Computer Engineering, Shiraz Branch, Azad University, Tabriz, Iran

³Faculty of Information Technology, Monash University, Clayton, Victoria, Australia

⁴Institute for Transport Studies, University of Leeds, Leeds, UK
Correspondence
 Mahdi Rezaei, Institute for Transport Studies, University of Leeds, Leeds, LS2 9JT, UK.
Email: m.rezaei@leeds.ac.uk
Funding information

Horizon 2020 Framework Programme, Grant/Award Number: 101006664

Abstract

Following the successful development of advanced driver assistance systems (ADAS), the current research directions focus on highly automated vehicles aiming at reducing human driving tasks, and extending the operational design domain, while maintaining a higher level of safety. Currently, there are high research demands in academia and industry to predict driver intention and understating driver readiness, e.g. in response to a “take-over request” when a transition from automated driving mode to human mode is needed. A driver intention prediction system can assess the driver’s readiness for a safe takeover transition. In this study, a novel deep neural network framework is developed by adopting and adapting the DenseNet, long short-term memory, attention, FlowNet2, and RAFT models to anticipate the driver maneuver intention. Using the public “Brain4Cars” dataset, the driver maneuver intention will be predicted up to 4 s in advance, before the commencement of the driver’s action. The driver intention prediction is assessed based on 1) in-cabin 2) out-cabin (road) and 3) both in-out cabin video data. Utilizing K -fold cross-validation, the performance of the model is evaluated using accuracy, precision, recall, and F1-score metrics. The experiments show the proposed DIPNet model outperforms the state-of-the-art in the majority of the driving scenarios.

1 | INTRODUCTION

According to a recent report by the World Health Organization (WHO), around 1.35 million people pass away annually in road accidents, globally [1]. The statistics only include fatalities of passengers due to car accidents [2]. Among the contributing factors, sudden maneuvers such as lane changes and turning play important roles in road accidents [3]. To reduce the number of such fatal accidents, a mechanism that can understand the driver’s intention before performing a dangerous maneuver can be helpful as an ADAS in preventing such fatal actions. Driver intention prediction can be also helpful in identifying the driver’s readiness for a safe takeover transition in L3 automated vehicles, based on the relevance of driver maneuver intention in accordance with the current driving scenario [4].

Over the past decade, many industrial and academic research studies have focused on developing autonomous vehicles (AVs). The AV systems have not yet fully covered the SAE Level 3 (L3) standard. There are still critical aspects of the systems that

need to be improved, such as developing more sophisticated data analytics capabilities, scene understanding [5], better collaboration tools, security of users’ data stored in the systems, as well as reducing computational costs for decision-making and predictive functions. The prediction horizon and response time required for L3 automated vehicles may vary from a few milliseconds to a few seconds depending on the automated driving function (ADF), the environment’s complexity, and driving scenarios. Short prediction horizons are used for lower-level tasks such as obstacle avoidance and recognition of driver intent, while longer horizons are needed for higher-level tasks such as pedestrian crossing intent prediction, or route planning. By proposing a model with a prediction horizon of up to 4 s in advance and real-time processing we resolve one of the main prerequisites of the L3 systems to a great extent.

Google and Tesla are among the leading industries that have made significant progress in autonomous vehicles [6]. Similarly, researchers in academia have extensively studied semi-autonomous assisted driving. As a result, the ADAS, SAE

This is an open access article under the terms of the [Creative Commons Attribution-NonCommercial License](https://creativecommons.org/licenses/by-nc/4.0/), which permits use, distribution and reproduction in any medium, provided the original work is properly cited and is not used for commercial purposes.

© 2023 The Authors. *IET Intelligent Transport Systems* published by John Wiley & Sons Ltd on behalf of The Institution of Engineering and Technology.

L3, and L4 vehicles, and cooperative automated driving via cooperative adaptive cruise control (CACC) [7] have led to promising perspectives to reduce traffic accidents, as well as reducing greenhouse gas emissions, resilient mobility, and the possibility of performing stress-free non-driving related tasks (NDRT) in automated mode [6, 8]. These systems are designed and equipped with sensory systems to understand information about road and driving conditions, assess hazards and driving warnings, and provide audio/visual requests to drivers [9]. Such systems aim at higher safety either by taking the driving control of the vehicle, by providing additional information to the human driver, or by identifying the driver's intention based on the characteristics of the driver's behavior and driving environment. It has been proven that with the help of advanced deep-learning techniques and computer vision, it is possible to predict the driver's intended maneuvers a few seconds in advance. Predicting maneuvers can be gained with a high level of accuracy by monitoring the driver's behavior inside the car (e.g. head pose, eye movement) and using the vehicle's dynamic information (e.g. speed, position) as well as environmental data (e.g. lanes configuration, presence of intersections, and position of the other road users) [3].

The majority of earlier studies in the field of driver maneuver prediction have mostly focused on extracting information from video frames of driver observations [2]. Numerous studies have demonstrated that driver behavior, particularly eye movements, can be employed to assure safe takeover behavior in conditionally automated vehicles [10] as well as for activity recognition [11, 12]. Other information such as head postures is also considered from video frames of driver observations in some literature [3, 6, 13–18]. In some of these works, a mixture of information such as road traffic information [19] or car dynamics status is fed to the model as external data. Refs. [6, 18] predict the driver's intention using in-cabin videos only. Ref. [6] uses head pose and eye movement features from driver observation videos. In ref. [18], in addition to head pose and eye movement features, the environment information is also fed to the model, manually. While the out-cabin video can be extremely informative (like traffic lights status, repairs on the road, and accident situations on the road) and transmit information that the interior video does not [2]. Some other works like refs. [3, 13–17, 20–22] apply both in and out-cabin videos and with the help of extra manual features predict the intention of the driver. However, adding extra information may not necessarily improve the AV's performance. In addition, manually extracted features are not applicable to practical use cases. Manual features also require a heavy load of processing and increase computational overload. Moreover, road traffic is too complicated for hand-crafting explicit features. Deep learning techniques have recently developed the domain, switching the recognition paradigm from manual feature descriptor development (e.g. body pose) to end-to-end learning of high-quality representations straightly from visual input employing convolutional neural networks (CNNs) [20]. Refs. [2] and [20] predict the driver's intention directly from both in and out-cabin videos without considering handcrafted and extra features (e.g. head

pose). Although the reported results in these two works are impressive, they can be improved more.

In this paper, we propose and design an end-to-end deep learning architecture based on in-cabin and out-cabin data to alleviate the aforementioned issues and challenges for the sake of coming up with a relatively accurate system for driver intention prediction. We also utilize both in-cabin and out-cabin videos effectively without using handcrafted (manual) features and detect the vehicle motion information from video data using emerging computer vision techniques, to enhance the results of driver's intention prediction. Since the system needs to be sufficiently fast to be applicable in such a real-world application, the low complexity of the model will be also prioritized, without sacrificing the model's accuracy. The proposed method is a data-driven approach and such approaches are becoming increasingly popular in automated driving as they do not require any prior knowledge of the physics behind the underlying system and can be used to accurately model highly-nonlinear behaviors. They also allow for more generalizability, as they learn from large datasets rather than being limited by assumptions associated with specific types of vehicle models. Furthermore, data-driven approaches are more flexible as they can easily incorporate new data sources and deal with variable sensor inaccuracies or occlusions. Finally, since data-driven models can generally train faster than their model-based counterparts, they require significantly lower computational resources.

The main contribution of this research can be summarized as follows.

- We develop a four-stream deep convolutional and recurrent neural network-based model. Different from the existing models for the intention prediction of the drivers, two streams perform based on the DenseNet-long short-term memory (LSTM) network to capture the features of the in-out cabin videos more efficiently. This processes the spatial and temporal features as a whole, which improves the prediction performance. The other two streams are built based on deep LSTM to extract the features of the optical flow by taking the in-out cabin video attributes into account since the intention prediction of each driver depends on a wide range of variables (e.g. traffic uncertainty). The proposed model has the ability to exploit data with high diversity (e.g. different weather conditions, and different types of roads in the city, or outside of the city).
- We integrate a global attention mechanism with the DenseNet-LSTM network to discriminate the importance of features and improve the performance of the proposed model.
- We apply an implicit semantic data augmentation algorithm (ISDA) to augment the dataset with semantically transformed samples and enhance prediction performance.
- We conduct experiments to validate the effectiveness of the proposed model. Evaluation results on the public dataset “*Brain4Cars?*” (<https://www.brain4cars.com>, and the mirror backup [link](#)) reveals that the proposed model achieves better

performance compared to other common models for driver intention prediction.

We provide further details in the next sections. The rest of the paper is organized as follows: In Section 2, the existing literature and state-of-the-art methods are concisely reviewed. The proposed method is described in Section 3, including the relevant datasets, details of the preprocessing, training procedures, and evaluation metrics. In Section 4, the results are presented and the proposed method is compared with related works. Eventually, the concluding remarks are listed in Section 5, along with further suggestions for future research directions.

2 | RELATED WORK

Multiple categories of related work in the driver maneuver intention prediction will be reviewed in two categories of data-based or model-based approaches. All related works reported in this section use Brain4Cars [14] dataset.

2.1 | Data-based approaches

Gite et al. [6] developed a driver's movement tracking (DMT) algorithm using only inside videos of the Brain4Cars dataset. A fusion of Spatio-temporal data points (STIPs) for DMT was introduced to improve the action anticipation performance, and a fast eye gaze algorithm to track eye movements were employed. Applying the F-RNN-DMT architecture they gained an accuracy of 96.21%, a precision rate of 94.11%, and a recall rate of 97.56%.

In ref. [18], Moussaid et al. employed two processing sections to predict the intended maneuver. The first section focused on feature extraction using a CNN DenseNet121 [23] architecture, followed by obtaining a data frame with 256 attributes combined with the exterior features. The proposed method was able to predict the turn/u-turn maneuver, 3.75 s in advance, with an accuracy of 94.1%.

In another attempt by Brains4Cars, they offered a sensory-fusion deep learning architecture based on recurrent neural networks (RNNs) with LSTM units called F-RNN-UL and F-RNN-EL to predict maneuvering on Brain4Cars dataset, using video data from both in-cabin and out-cabin data including facial landmarks, head pose, car speed, GPS information, and lane configuration.

Their sensory fusion deep learning approach obtained a precision and recall rate of 84.5% and 77.1%, respectively. The model is able to anticipate the maneuvers 3.5 s (on average) before they happen. Combining multiple sensory streams, the precision and recall rate improved to 90.5% and 87.4%, respectively [16].

In ref. [21], the proposed prediction system utilized a deep bidirectional recurrent neural network (DBRNN). They used both in/out data and evaluated the performance of the system for braking, lane change, and turning anomaly action prediction on their suggested data and Brain4Cars dataset. The

research reports an average accuracy of 80% within 3 s from the braking event.

Rekabdar et al. [22] proposed a novel deep learning architecture and utilized sensory data sources such as GPS location, car speed, and visual data from the camera installed inside and outside the car, and other related car sensors presented on the Brain4Cars dataset. The proposed method introduced a sensor-fusion deep learning framework using a combination of dilated CNN and CNN max-pooling pairs. The results of precision and recall were reported as 91.8% and 92.5%, respectively.

Zhou et al. [15] presented a cognitive fusion recurrent neural networks (CF-RNN) model based on the cognition-driven model and data-driven model. CF-RNN includes two LSTM units that cognitively fuse in-cabin and out-cabin videos. The outputs of the two LSTM units were adjusted by the human cognition time process, which led to an F1-score improvement from 88.9% to 92.1% on the Brain4Cars dataset.

Inspired by ref. [4], the authors of ref. [13] presented an architecture based on RNN and LSTM, using the Brain4Cars dataset that combines both information from inside and outside the car to predict the driver's actions, which leads to achieving 92.12%, 87.95%, 95.95%, and 86.1% for accuracy, precision, recall, and F1-score, respectively.

Zhou et al. [17] introduced a CF-LSTM model based on a cognition-driven method and a data-driven method inspired by ref. [16] for feature extraction. This model includes two LSTM units for both interior and external currents of the car, which describes the external features including speed, the lane configuration, internal features, driver head movement, and driver's face landmark using the CLM or CLNF algorithm [14]. The authors also introduce an architecture called the predictive-Bi-LSTM-CRF model and a comprehensive evaluation metric that predicts the maneuver with an accuracy of 94.83% and the F1-score of 93.6% on the Brain4Cars database.

Compared to the previous works, refs. [20] and [2] experimentally validate that both in/outside videos contain complementary information and do not use manual information.

In ref. [20] the authors propose a model to anticipate the driver maneuver intention directly from videos in an end-to-end method. The proposed model consists of three components: a FlowNet [24] architecture for optical flow extraction to obtain the motion-based representations, a 3D residual network (3D ResNet) for maneuver classification, and an LSTM unit for handling temporal data of varying lengths. They fused driver observation data from inside and outside the cabin and fine-tuned the proposed model based on a pre-training on the large-scale Kinetics dataset, resulting in an accuracy rate of 83.12% and an F1-score of 81.74% on the Brain4Cars dataset.

The architecture proposed in ref. [2] utilizes two data streams: outside and inside frames of the car. First, using FlowNet 2.0 [25], the optical flow of images is generated from the main out-cabin frames and fed to a ConvLSTM encoder part of the model.

In another attempt, using a 3D ResNet-50 network, feature extraction was performed from the in-cabin frames, resulting in an accuracy of 83.98% and the F1-score of 84.3% on the Brain4Cars dataset.

TABLE 1 The summary of the performance of related works on driver intention prediction including single-data modality approaches (refs. [2] and [20]) and multi-modal feature-fusion based approaches (refs. [6, 13]–[18, 21], and [22]).

References	Data Source	Method	Accuracy	Precision	Recall	F1-score	PH (s)
Rong et al. [2]	Inside	ConvLSTM auto-encoder & ResNet50	77.4%	–	–	75.5%	0
	Outside		60.9%	–	–	66.4%	
	In-out		84.0%	–	–	84.3%	
Zhou et al. [17]	In-out	CF-LSTM and predictive-Bi-LSTM-CRF	–	92.4%	94.7%	93.6%	–4.10
Gite et al. [6]	Inside	F-RNN-DMT	96.2%	94.1%	97.6%	–	0
Moussaid et al. [18]	In-out	CNN-LSTM	94.1%	–	–	–	–3.75
Gite et al. [13]	In-out	RNN-LSTM	92.1%	88.0%	96.0%	–	0
Gebert et al. [20]	Inside	3D ResNet & LSTM using FlowNet 2	83.1%	–	–	81.7%	0
	Outside		53.2%	–	–	43.4%	
	In-out		75.5%	–	–	73.2%	
Tonutti et al. [3]	In-out	DA-RNN & LSTM-GRU	–	92.3%	90.8%	91.3%	–4
Zhou et al. [15]	In-out	CF-RNN	–	92%	92.3%	92.1%	–3.30
Rekabdar et al. [22]	In-out	Dilated CNN	–	91.8%	92.5%	–	–3.76
Olabiyi et al. [21]	In-out	DBRNN - breaking	80.0%	–	–	–	–2
		DBRNN - change	80.0%	–	–	–	
		DBRNN - turning	90.0%	–	–	–	
Jain et al. [16]	In-out	F-RNN-UL & F-RNN-EL	–	90.5%	87.4%	–	–3.58
Jain et al. [14]	In-out	AIO-HMM	–	77.4%	71.2%	80.0+%	–3.5

Overall, the data-driven approaches are efficient and cost-effective when it comes to modeling large datasets with high prediction accuracy. Furthermore, they do not require manual interventions or feature engineering efforts such as those required in traditional AI models. On the other hand, these approaches may not always be able to accurately explain complex interactions between features and output or capture changes in user preferences over time which require more sophisticated modeling techniques.

2.2 | Model-based approaches

Many works including refs. [3, 13–17, 21, 22] as articles that use out-cabin videos in their framework in addition to driver observation videos. They also use additional auxiliary features (e.g. head pose, eye movement, vehicles' dynamic information, and environment data) along with the in-cabin videos.

Refs. [6] and [18], are among the articles that only pay attention to the driver's observation video (in-cabin video) and ignore the outside of the cabin. The models try to predict the driver's intention, using visual features such as head pose, eye movement, vehicles' dynamic information (e.g. speed, position), as well as online environmental data (e.g. lanes configuration and the positions of intersections).

The Brain4Cars [14] research group was among the teams in the field that published one of the first naturalistic driving datasets. As one of the earliest research, Jian et al. [14] (from the Brain4Cars team) proposed a model called auto-regressive input–output HMM (AIO-HMM) by utilization of a mixture of

in/out cabin data such as driver's heave movement tracking [26] and car speed, which enabled the model to anticipate the next maneuver 3.5 s prior it occurs, with an F1-score rate of over 80%.

Tonutti et al. [3] applied an LSTM-GRU model for feature extraction and maneuver prediction using the driver's head features, eye movement, and driving environment along with domain-adversarial RNN (DA-RNN) model for domain adversarial training to achieve domain adaptation. DA-RNN model was evaluated on the Brain4Cars dataset as the source domain and a proposed new dataset [27] which includes 113 videos as the target domain.

Moreover, to optimize the extraction of domain-independent features, a fine-tuning method was employed, which led to precision, recall, and F1-score of 92.3%, 90.8%, and 91.3%, respectively on the Brain4Cars dataset. Similar results of 89.4%, 92.2%, and 90.8%, were reported on the proposed dataset.

Model-based approaches provide a more rigorous approach to understanding the complex dynamics of a vehicle's behavior by generating models that accurately simulate and predict driver outcomes. While data-driven approaches are able to process large amounts of data quickly and directly, creating predictive models based on observed patterns in the data. Furthermore, neural networks can be employed to further enhance the performance of both methods. In conclusion, both model-based and data-driven approaches for predicting driver intention on the Brain4cars dataset offer unique advantages. Choosing an applicable approach ultimately depends on the specific requirements and scope of a given study, as well as the size and availability of an appropriate dataset.

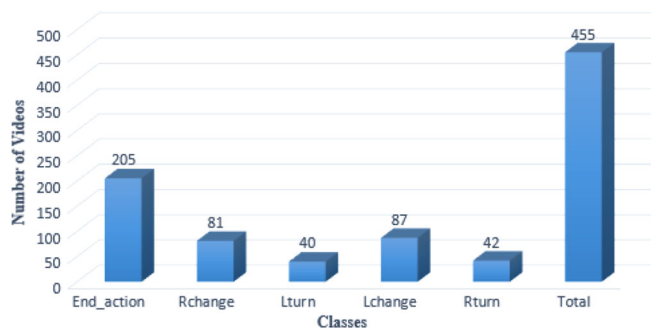


FIGURE 1 Distribution of each class in Brain4Cars dataset.

Table 1 represents a summary of our literature on a diverse set of 12 related works in the field and their performances. Some of the reviewed works are based on single data modality (similar to our approach) and some are implemented based on multi-modal feature fusion strategies [28] (e.g. driver's head pose features, eye movement, vehicle's lane, etc.) which are beyond the scope of this research. The table summarizes the core methodology, the performance of each model, and the prediction horizon (PH). The PH represents the prediction of the driver's intention, t s in advance, prior to the driver's actual maneuver taking place. In this research, we only aim to improve the accuracy and speed of single-modality-based approaches which in turn may help the enhancement of multi-modal fusion-based approaches, as well.

3 | METHODOLOGY

Most of the reviewed research work has neglected the limited size and unbalanced labeling of the Brain4Cars datasets and their effects on the training of their models. Figure 1 illustrates the number of videos in each class of the Brain4Cars dataset. Due to the relatively small size of the dataset, deep learning-based models may face overfitting issues. Furthermore, as can be noticed one of the other weaknesses of this dataset is the unbalanced number of videos in each label (from 40 to 205) which may prevent the generalizations of the model.

Spatial information, as well as temporal information, are critical for predicting driver behavior. Inspired by Simonyan et al.'s work [29] and employing temporal information, we propose a framework consisting of a convolutional neural network and LSTM network to predict driver's behavior and tackle the above-mentioned challenges.

The network consists of three main parts: a spatial stream including the DenseNet module, a temporal stream consisting of two successive LSTMs, and a classification part that is responsible for classifying extracted features, as shown in Figure 2. We used transformer learning in the spatial part by employing the DenseNet121 model pre-trained on the ImageNet dataset followed by drop block, average pooling, LSTM, and global attention block. As a result, spatial features were obtained. The DenseNet architecture was utilized to extract features. Compared to existing state-of-the-art alternatives, DenseNet has fewer parameters while still being deep

enough to capture efficient features, making them more suitable for devising AVs.

They usually have 60 000 to 70 000 parameters, being considerably smaller than the number of ResNet architectures parameters containing about 25 to 26 million parameters. The number of DenseNet parameters in the proposed architecture is 10 000.

Motion analysis is also one of the most fundamental and challenging problems in machine vision that can be widely used in various applications, such as automatic driving, performance detection, scene perception, and robotics [30]. Recurrent all-pairs field transforms (RAFT) [31] and FlowNet2 [25] models are utilized in our model to extract the required optical flows for the in-car and out-of-car videos, respectively. We utilized these models as they are exclusively designed to produce optical flow as spatial pyramid networks and to gain high efficiency and accuracy, small model size, and low execution time in practical applications [32].

In the temporal part in Figure 2, to extract the temporal feature, instead of using 3D ConvNet [33, 34] or 2D ConvNets+LSTM [35, 36] methods having many parameters which increase the probability of overfitting due to the small number of data, two layer of LSTM were used.

In the spatio-temporal module, we used two LSTM layers to extract temporal features, rather than using 3D ConvNet as in refs. [33, 34] or 2D ConvNets+LSTM as in refs. [35, 36]. This will highly reduce our model parameters and consequently decreases the likelihood of overfitting. The first LSTM layer is responsible for extracting the general features such as vertical or horizontal movements in each frame. The second one extracts the desired temporal features, the most relevant and informative features regarding the driver's behavior anticipation, from the set of input frames. The classification module consists of two fully connected layers and one dropout, which classifies the driver behavior into five categories, after concatenation of the optical flow features and the spatio-temporal features.

3.1 | Dataset

In this study, The Brain4Cars [14] dataset was utilized for evaluating the proposed model. Brain4Cars dataset includes two different views videos: (a) driver observation videos (1088 px \times 1920 px, 30 fps) and (b) out-cabin videos (480 px \times 720 px, 30 fps) which are recorded simultaneously and synchronized [2].

There are five classes of maneuvers in the dataset: go straight, left lane change, left turn, right lane change, and right turn. According to the Brain4Cars dataset, all videos only include the driver's behavior before the actual maneuver occurs, i.e. no maneuver is performed during the video.

This dataset has been collected from 10 drivers with a car equipped with a camera, and the videos are annotated for 700 events in total, containing 274 lane changes, 131 turns, and 295 randomly sampled instances of driving straight [2].

In this study, 80% of the short videos are used for training (70% for training the model parameters, and 10% for validation

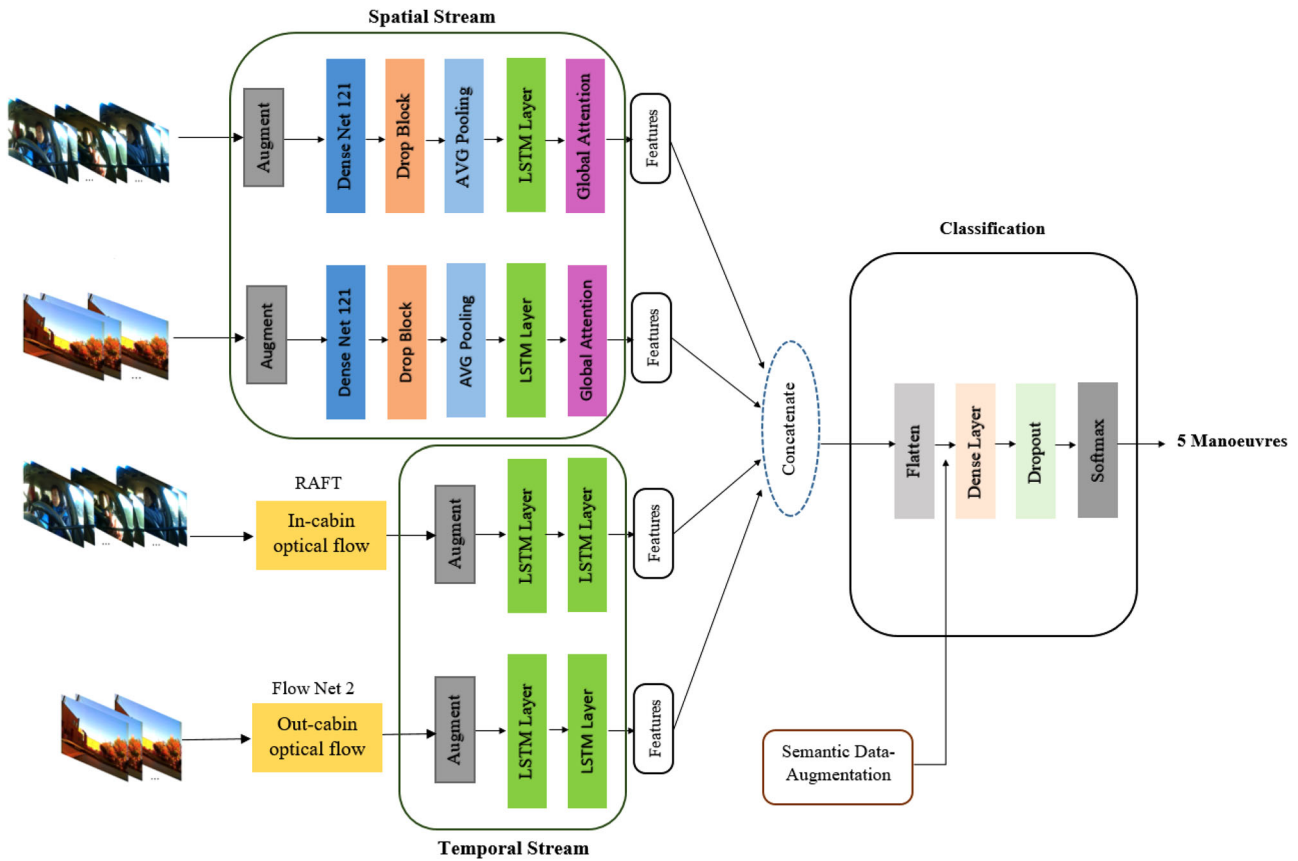


FIGURE 2 Schematic illustration of the proposed architecture. In the first scenario, the branches using an outside view are eliminated, and in the second scenario, the branches using an inside view are ignored.

with the purpose of generalizability analysis) and 20% of the short videos are used as the test set. Performance metrics are calculated for the experiment from this set to test the final performance of the DIPNet, with the length of 5 s, which are the crucial and golden seconds for a decision-making system. This can be also an important stage for transferring the vehicle's control from an automated mode to a human mode and vice versa.

3.2 | Model architecture

The proposed model in Figure 2 consists of 4 input sources: the main input video frames from inside and outside of the cabin as well as optical flow frames from inside and outside of the cabin. For each of the input sources, we select one frame out of 10 frames, so that each $5\text{-s} \times 30\text{ fps}$ video would consist of 15 frames.

Regardless of the dataset and video length, the proposed DIPNet model is specifically designed to handle complex data inputs, so the model is capable of processing and analyzing various datasets with different spatio-temporal video lengths. The DIPNet framework can increase the prediction time by more than 4 s and can be also adapted to any dataset with a length

of longer than 4 s with no limitations. Although, the maximum possible time also depends on the hardware specification.

3.2.1 | Pre-processing and data augmentation

As the pre-processing step, all inputs are resized to 128×128 pixel. Then a data augmentation is applied to the raw images in the first and second branches. Data augmentation is also applied to the output of RAFT and FlowNet2, in the third and fourth branches. (Figure 2). This includes translation, flip-left-to-right (flipLR), cutout [37], and a technique called Augmix as in ref. [38]. As part of the translation Augmentation, the raw and optical flow image was moved by 4 pixels in both directions. In addition, flipLR, which flips the image vertically, is applied, and due to its impact on its label, the label also changed. For example, a turning left label will be changed to turning right after applying a flipLR augmentation. Another augmentation technique named cutout is a regularization technique that randomly masks out square regions. Augmix also is a data processing technique, which mixes randomly generated augmentations (auto contrast, equalize, posterize, solarize). More details and information are addressed in ref. [38]. Some samples of augmented images have been depicted in Figure 3.

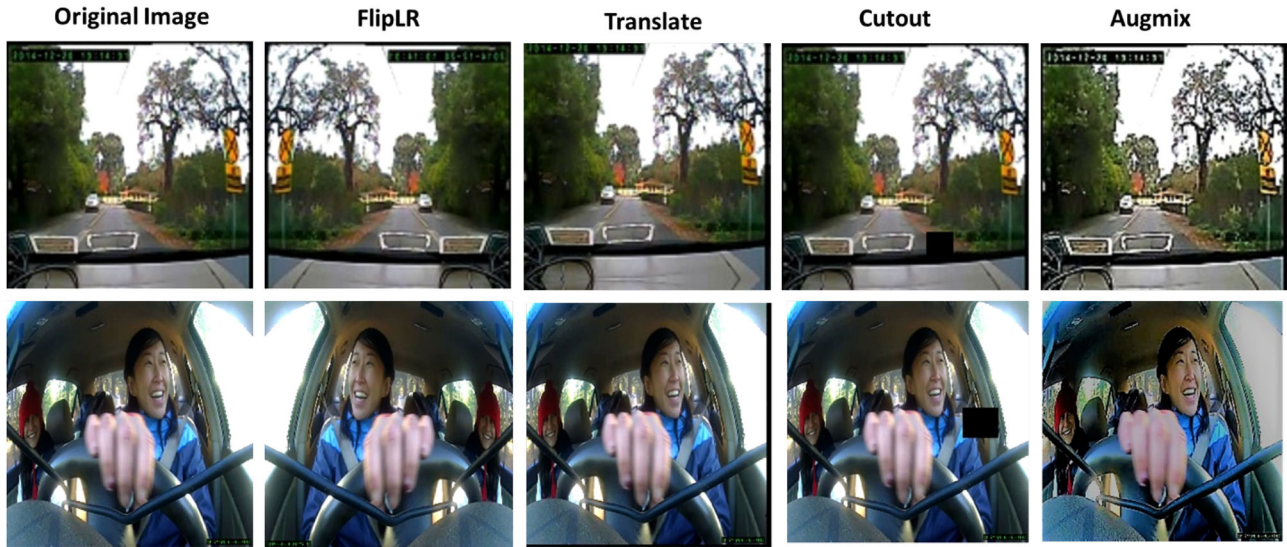


FIGURE 3 Some samples of applied augmentations (augmix, cutout, translate). Each row illustrates a sample image, and the applied augmentation is shown on top of each column.

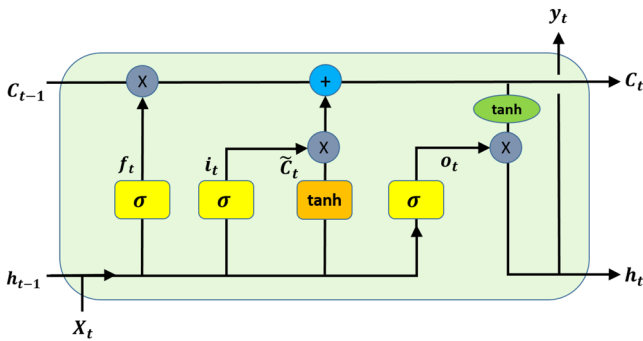


FIGURE 4 LSTM cell with its internal structure [41].

3.2.2 | Feature extraction

Following the augmentation phase, the in-cabin and out-cabin images are fed to the DenseNet121 model [39] to extract the features. The extracted features with the size of 1024 are then passed through a Dropblock layer [40] with a block size of 5. An AVGPooling layer is then added to the Dropblock output, followed by an LSTM layer with 512 memory units and a Global Attention layer. We utilized LSTM as it remembers the previous information in time series data. As depicted in the schematic of LSTM cell [41] (Figure 4), the LSTM consists of an input gate (i), forget gate (f), output gate (o), and cell state (c). The function of the input gate is storing and updating input information in the current state and the forget gate’s mission is to decide either to forget or to retain the previous data. The output gate is the output of the network and the memory cell state (c) stores long-term information.

The LSTM network is expressed as an artificial neural network (ANN) where the input vector $x = (x_1, x_2, x_3, \dots, x_t)$ at timestamp t , maps to the output vector $y = (y_1, y_2, \dots, y_t)$,

through the calculation of i_t , f_t , and o_t which represent input gate, forget gate and output gate. We define the output of the LSTM gate at timestamp t as follows:

$$i_t = \sigma(W_i \cdot [b_{t-1}, x_t] + b_i) \tag{1}$$

where W , σ , x_t , and b_{t-1} are weight matrix, Sigmoid activation function, input vector at time t , and output (or hidden state) vector of the previous LSTM cell (at time $t - 1$), respectively. b_i is the Bias vector.

Similarly, the output of the forget gate and output gate can be represented as follows:

$$f_t = \sigma(W_f \cdot [b_{t-1}, x_t] + b_f) \tag{2}$$

$$o_t = \sigma(W_o \cdot [b_{t-1}, x_t] + b_o) \tag{3}$$

The cell state (c_t), candidate for cell state at timestamp t (\tilde{c}_t), and the final output (h_t) are defined as follows:

$$c_t = f_t * c_{t-1} + i_t * \tilde{c}_t \tag{4}$$

$$\tilde{c}_t = \tanh(W_c \cdot [b_{t-1}, x_t] + b_c) \tag{5}$$

$$h_t = o_t * \tanh(c_t) \tag{6}$$

To get the memory vector for the current timestamp (c_t) the candidate is calculated. Using the above equations, the cell state knows what should be forgotten from the previous state (i.e. $f_t * c_{t-1}$) and what should be considered from the current timestamp (i.e. $i_t * \tilde{c}_t$). (note: * represents the element-wise multiplication of the vectors.)

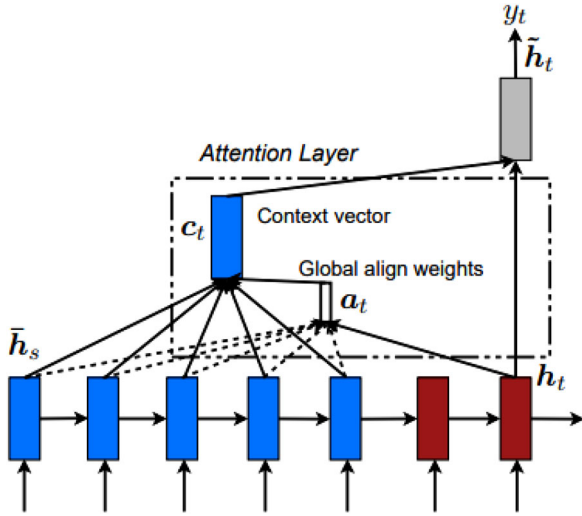


FIGURE 5 Schematic architecture of global attention [42]. Blue and red blocks are shown encoding and decoding phases, respectively.

Lastly, the cell state is filtered and passed through the activation function which predicts what portion should appear as the output of the current LSTM unit at timestamp t . We can pass this b_t the output from the current LSTM cell through the Softmax layer to get the predicted output (y_t) from the current cell.

In the global attention module, generally, we can consider three types of attention: 1) global and local attention (local-m, local-p), 2) hard and soft attention, and 3) self-attention. Unlike local attention, the global attention derives a context vector based on all hidden states of the LSTM, to the entire input state space.

As can be seen from Figure 5, at the decoding phase, at each time step t , the hidden state b_t is taken as input at the top layer of a stacking LSTM. By comparing the current target hidden state b_t with each source hidden state b_s , a variable-length alignment vector a_t is derived, where its size equals the number of time steps on the source side as follows [42]:

$$a_t(s) = \text{align}(b_t, \vec{b}_s) = \frac{\exp(\text{score}(b_t, \vec{b}_s))}{\sum_s \exp(\text{score}(b_t, \vec{b}_s))} \quad (7)$$

where $\exp()$ refers to the exponential function, and the score is referred to as a content-based function for which we consider three different alternatives:

$$\text{score}(h_t, \vec{h}_s) = \begin{cases} h_t^\top \vec{h}_s & \text{dot} \\ h_t^\top W a \vec{h}_s & \text{general} \\ v a^\top \tanh(W a [h_t; \vec{h}_s]) & \text{concat} \end{cases}$$

Given the alignment vector as weights, the context vector c_t is computed as the weighted average over all the source hidden states.

In order to predict the current target feature y_t , a context vector c_t that captures the relevant source-side information, should be derived. Then a simple concatenation layer is employed to incorporate the target hidden state b_t and the source-side context vector c_t to produce an attentional hidden state as follows:

$$\vec{b}_t = \tanh(W_c [c_t; b_t]) \quad (8)$$

The attentional vector \vec{b}_t is then fed through the Softmax layer to produce the predictive distribution formulated as:

$$p(y_t | y_{<t}, x) = \text{softmax}(W_s \vec{b}_t) \quad (9)$$

where W_c and W_s are weight matrices to be learned in the alignment model.

Then the in-cabin and out-cabin optical flow frames which are produced by recurrent all-pairs field transforms (RAFT) (in the third branch) and FlowNet2 [25] (in the fourth branch) are augmented by translation and FlipLR techniques. The produced data is then resized to 128×384 and are fed into two successive LSTM layers with 128 memory cells.

In the first scenario, when the only inside view is utilized, the second and fourth branches of the model are eliminated. Similarly, when we use the outside view in the second scenario, the first and third branches are ignored.

3.2.3 | Classification

In the second part of the model, the extracted features are classified. Firstly, the extracted features from the four input branches are concatenated and passed through a flattened layer. Then the implicit semantic data augmentation algorithm (ISDA) [43], a novel technique for augmentation, is applied, as seen in Figure 6.

In the next step, the output of the semantic data augmentation module is passed through a dense layer with 512 neurons, followed by a dropout layer with a rate of 0.45. Eventually, the Softmax layer is employed, and the probability of the given input is generated. ISDA is a novel algorithm that unlike previous augmentation algorithms changes the context of an image semantically. More specifically, consider $D = (x_i, y_i)$ is a training set, where $y_i \in 1, \dots, C$ is the label of the i th sample x_i over C classes and G and Θ are deep network and its weights, respectively. $a_i = [a_{i1}, \dots, a_{iA}]^T = G(x_i, \Theta)$ as an A dimensional vector shows the deep feature of x_i , and a_{ij} denote the j th element of a_i . a zero-mean multi-variate normal distribution $N(0, \sum y_i)$ is established to achieve semantic directions to augment a_i , where $\sum y_i$ depicts the class-conditional covariance matrix. The algorithm for the covariance matrices is as follows:

$$\mu_j^{(t)} = \frac{n_j^{(t-1)} \mu_j^{(t-1)} + m_j^{(t)} \mu_j^{(t)}}{n_j^{(t-1)} + m_j^{(t)}} \quad (10)$$

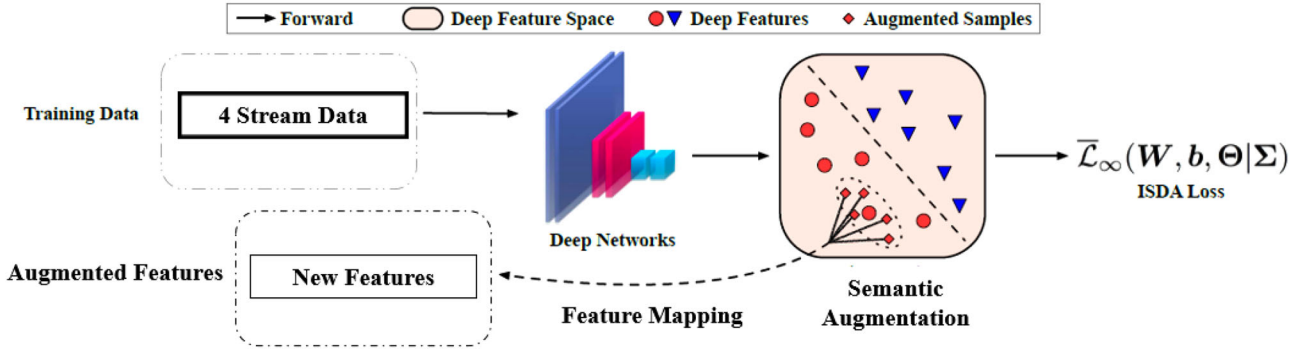


FIGURE 6 The structure of the feature augmentation.

$$\Sigma_j^{(t)} = \frac{n_j^{(t-1)} \Sigma_j^{(t-1)} + m_j^{(t)} \Sigma_j'^{(t)}}{n_j^{(t-1)} + m_j^{(t)}} + \frac{n_j^{(t-1)} m_j^{(t)} (\mu_j^{(t-1)} - \mu_j'^{(t)}) (\mu_j^{(t-1)} - \mu_j'^{(t)})^T}{(n_j^{(t-1)} + m_j^{(t)})^2} \quad (11)$$

$$n_j^{(t)} = n_j^{(t-1)} + m_j^{(t)} \quad (12)$$

The estimations of average values and covariance matrices of the features of j th class at t th step are denoted by $\mu_j^{(t)}$ and $\Sigma_j^{(t)}$. $\mu_j^{(t)}$ and $\Sigma_j^{(t)}$ are the average values and covariance matrices of the features of j th class in t th mini-batch, respectively. The number of training samples in j th class only in t th mini-batch are denoted by $m_j^{(t)}$ and the total number of training samples which are in j th class in all t mini-batches are shown by $n_j^{(t)}$. To read more details of ISDA refer to ref. [43].

3.2.4 | Training

There is no official data split for training/validation/test parts of the Brain4Cars dataset, and each article has taken a different part randomly. However, in our fair comparison, we compared our model with related works that have exactly used the same data split proportions (i.e. 70%, 10%, 20%) for the training, validation, and test, respectively. Also, same as the compared works we used K -fold validation with $K = 5$ to show the independence of our approach to the train/test splits. Based on the 5-fold validation results, the model shows a very good generalization when we use different train/test parts of the same size.

In our work, three different scenarios are defined to predict the driver's actions and assess the performance of the proposed method. In the first scenario, the proposed model is trained with only inside-view images (in-cabin). In the second scenario, the proposed model is trained with only outside view images (out-cabin), and in the third scenario, the proposed model is trained with both inside and outside view images (in-out cabin).

The number of epochs was 320 with a batch size of 5 and an Adam [44] optimizer. The initial learning rate of 0.0003 was served, and the pattern of changing learning rate during training is depicted in Figure 7. The network was trained using the categorical cross-entropy loss function, and the training process was conducted on the Google Colab graphics processing unit.

In our proposed model, the DenseNet architecture is used for feature extraction, which has fewer parameters than other models reviewed in the literature review. As a result, the proposed method leads to lower computational costs and hardware requirements in the training and testing phase. The training time of the proposed model with a batch size 8, on a Google Colab platform with an NVIDIA T4 GPU is 45 s per epoch. The run time of the model in the test phase is however 180 ms (0.18 s) only, which is fast enough for the intended real-time application.

3.2.5 | Performance evaluation

To evaluate the performance of the proposed network like ref. [6], we used four standard performance metrics, described in Equations (13)–(16): accuracy, precision, recall rate, F1-score, as well as a confusion matrix. The elements to calculate the

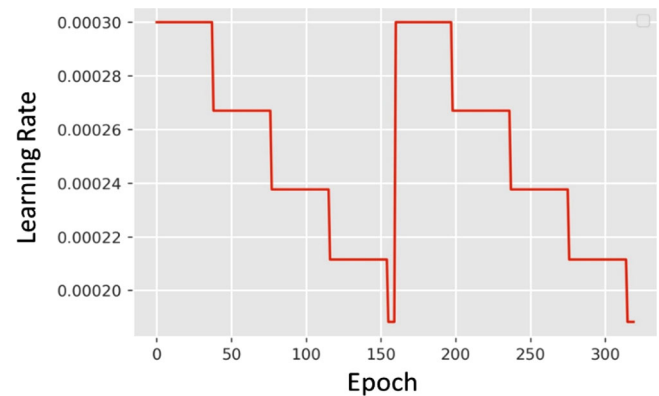


FIGURE 7 Learning rate scheduler during the training of the model.

TABLE 2 Performance of the model on the “inside view” dataset.

PH (s)	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
0 s	89.01	89.13	89.01	88.89
-1 s	83.51	83.22	83.52	83.20
-2 s	75.82	75.41	75.82	75.42
-3 s	60.43	56.38	60.44	56.41
-4 s	46.15	45.83	46.15	45.20

TABLE 3 Performance comparison of the proposed model with two other state-of-the-art models on the “inside view” video.

References	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)	PH (s)
Rong et al. [2]	77.40	N/A	N/A	75.49	0
Gebert et al. [20]	83.1	N/A	N/A	81.7	0
Ours	89.01	89.13	89.01	88.89	0

mentioned metrics are true positive (TP), true negative (TN), false positive (FP), and false negative (FN), which are defined as follows for the driver action prediction:

- True positive (TP) = correct action prediction.
- False positive (FP) = incorrect action prediction.
- True negative (TN) = predicting no action (i.e. driving straight) and the driver also does not perform any action and drives straight.
- False negative (FN) = predicting straight driving, but the driver performs an action.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (13)$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (14)$$

$$F1 - \text{score} = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \quad (15)$$

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \quad (16)$$

4 | RESULTS AND DISCUSSION

In this section, the performance of the proposed method is discussed for three different scenarios (in-cabin, out-cabin, and both in/out-cabin). The results of early detection time capability are provided (in seconds) for each scenario and listed in Tables 2, 5, and 8. The action is recognized t seconds before a maneuver take place i.e. $t \in (-4 \text{ s}, -3 \text{ s}, -2 \text{ s}, -1 \text{ s}, 0 \text{ s})$. Furthermore, the obtained results are compared with the state-of-the-art studies shown in Tables 3, 6, and 9. The 5-fold cross-validation for all experiments is provided in Tables 4, 7,

and 10. It should be noted that all the provided results are in percentage. Finally, the effect of the aforementioned augmentations in the performance of the system using the in/out cabin views as well as the confusion matrix for all scenarios are depicted.

It is worth mentioning that the original motivation for using K -fold cross-validation is to minimize the variance of the estimated performance of a learning algorithm by reusing the different subsets of the data for testing and training. K -fold cross-validation introduces randomness into the model evaluation process, and it forces a learning algorithm to train the model multiple times on different subsets of the data. This randomization reduces both the bias and variance of the estimated performance of the learning algorithm as illustrated in Table 10.

4.1 | In-cabin action recognition

As the first scenario, in this section, only the in-cabin images are utilized to predict the driver actions listed in Figure 3, the second row. Table 2 shows the accuracy, precision, recall, and F1-score as our evaluation metrics. As can be seen, the performance of the model increases from the $t = -4 \text{ s}$ towards the $t = 0 \text{ s}$ before the real action, and eventually reaches 89.01%, 89.13%, 89.01%, and 88.89%, respectively. In Table 3, the proposed work is compared with a couple of current studies including Rong et al. [2] and Gebert et al. [20] which use the same in-cabin data only. The results confirm our model outperforms the other works. In addition, we employed 5-fold cross-validation to guarantee that the distribution of training and test data is logical. Table 4 provides the accuracy rate of each step in the K fold cross-validation method in which K is equal to 5. In the OTC method which stands for “original, translate, and cutout”, the original image and the augmented image (by translate and cutout) are given to the model separately, and the best performance is selected as the final result. The mean of all steps, as well as the standard deviation, is listed in the last column. Similar to Table 2, the accuracy of the model in time 0 s is the highest (88%) and it fluctuates to 46%, 60%, 75%, 83% for times -4, -3, -2, -1 s, respectively. Utilizing OTC shows a performance improvement of 1–2% for each time step.

4.2 | Out-cabin action recognition

In the second scenario, the model was trained with only outside-view images. Following the same pattern in Table 2, the performance upsurges when approaching 0 s represented in Table 5. In 4 s before real action, the accuracy precision, recall, and F1-score were 42.86%, 44.18%, 42.86%, and 42.84%. In each step, they soared and reached 82.41%, 82.28%, 82.42%, and 82.24% in time = 0 s. Table 6 compared our model with state-of-the-art models that used only outside-view images, observing surpassing other works. Our model’s accuracy, precision, recall, and F1-score are 82.41%, 82.28%, 82.42%, and 82.24%, while the ref. [2] achieved an accuracy of 60.87% and

TABLE 4 Accuracy results of K -fold method using “inside view”, $K = 5$.

PH (S)	Method	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Mean \pm std (%)
0 s	Our	87.91	87.91	89.01	87.91	89.01	88.35 \pm 0.6
0 s	Our + OTC	87.91	87.91	89.01	89.01	89.01	88.57 \pm 0.6
-1 s	Our	80.21	83.51	83.51	81.31	83.51	82.41 \pm 1.5
-1 s	Our + OTC	83.51	83.51	83.51	81.31	83.51	83.07 \pm 0.98
-2 s	Our	75.82	74.72	73.62	74.72	75.82	74.94 \pm 0.92
-2 s	Our + OTC	75.82	74.72	74.72	75.82	75.82	75.38 \pm 0.60
-3 s	Our	59.34	60.43	58.24	59.34	60.43	59.56 \pm 0.91
-3 s	Our + OTC	60.43	61.53	60.43	59.34	60.43	60.43 \pm 0.77
-4 s	Our	43.95	45.05	45.05	43.95	46.15	44.83 \pm 0.92
-4 s	Our + OTC	48.35	45.05	46.15	46.15	46.15	46.37 \pm 1.2

TABLE 5 Performance of the model on the “outside view” dataset.

PH (s)	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
0 s	82.41	82.28	82.42	82.24
-1 s	76.92	76.65	76.92	76.64
-2 s	67.03	64.20	67.03	64.28
-3 s	56.04	51.04	56.04	51.50
-4 s	42.86	44.18	42.86	42.84

TABLE 6 Performance comparison of the proposed model with two other state-of-the-art models on the “outside view” video.

References	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)	PH (s)
Rong et al.[2]	60.87	N/A	N/A	66.38	0
Gebert et al.[20]	53.2	N/A	N/A	43.4	0
Ours	82.41	82.28	82.42	82.24	0

F1-score of 66.38%, and [20] obtained an accuracy of 53.2% and F1-score of 43.4%. Similarly, to calculate accuracy, the K -fold cross-validation method is also applied, which is shown in Table 7. The parameter K is equal to 5, and the OTC method described in Section 4.1 was utilized. Using OTC, the accuracy of 82.19% was obtained In time = 0 s, which is close to what was achieved in Table 5, 82.24%. The obtained results used only inside view in terms of precision, recall, F1-score, and accuracy.

4.3 | In-cabin and out-cabin action recognition

In this scenario, we utilized in-cabin and out-cabin images for training the model, resulting in the best performance comparing the two previous scenarios (in-cabin and out-cabin). Table 8 shows the accuracy of 98.90%, the precision of 98.96%, recall of 98.90%, and F1-score of 98.88% of this scenario in real-

time action. In Table 9, we compared our result with two other rival works [2, 20], and it can be seen that these present studies use both inside and outside views without manual features to enhance the performance as well. Our results were 98.90%, 98.96%, 98.90%, and 98.88% for accuracy, precision, recall, and F1-score, outperforming other rival methods listed in Table 9. Table 10 also provides a K -fold cross-validation method similar to previous scenarios. In time = 0 s, we achieved an accuracy of 98.46%, which is close to obtained result in Table 8. As aforementioned, OTC is served to enhance the performance result. Finally, Table 11 illustrates the recognition results of five classes of maneuvers under different prediction horizons (-4, -3, -2, -1, 0 s) when using both “inside view” and “outside view” videos.

4.4 | Effect of augmentation on performance

Different augmentation methods including flipLR, translate, cutout, and augmix were applied. Figure 8a–d illustrates the effect of the augmentation on the accuracy, precision, recall, and F1-score, respectively. Plot A shows the model without employing any augmentation. In plot B, FlipLR is added to A. Adding a cutout to configuration B, lead to plot C which surpassed the previous ones (A and B). Similarly, plot D shows the results of augmix augmentation added to C. Plot E, represents the application of all augmentations (FlipLR, cutout, augmix, and translate), which excels all other explained methods in terms of the four metrics, as seen in Figure 8.

4.5 | Confusion matrix

Moreover, to represent the performance of the proposed architecture, the confusion matrixes are depicted in Figure 9, where the numbers in diagonal stand for the count of correctly recognized samples from corresponding classes. It is observable that the model error in the confusion matrix (c) shows the model trained with both in-cabin and out-cabin views is less than the other two matrixes (a) and (b) which represent the model trained

TABLE 7 Accuracy results of K -fold method using only “outside” view dataset, $K = 5$.

PH (s)	Method	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Mean \pm std (%)
0 s	Our	80.21	81.31	81.31	82.41	82.41	81.53 \pm 0.92
0 s	Our + OTC	82.41	82.41	81.31	82.41	82.41	82.19 \pm 0.49
-1 s	Our	75.82	76.92	75.82	75.82	76.92	76.26 \pm 0.60
-1 s	Our + OTC	75.82	76.92	76.92	75.82	76.92	76.48 \pm 0.60
-2 s	Our	65.93	65.93	64.83	65.93	67.03	65.93 \pm 0.77
-2 s	Our + OTC	67.03	65.93	67.03	68.13	67.03	67.03 \pm 0.77
-3 s	Our	54.94	53.84	56.04	56.04	56.04	55.38 \pm 0.98
-3 s	Our + OTC	56.04	56.04	56.04	57.14	56.04	56.26 \pm 0.49
-4 s	Our	41.75	41.75	40.65	41.75	42.86	41.75 \pm 0.78
-4 s	Our + OTC	41.75	42.86	42.86	43.95	42.86	42.86 \pm 0.77

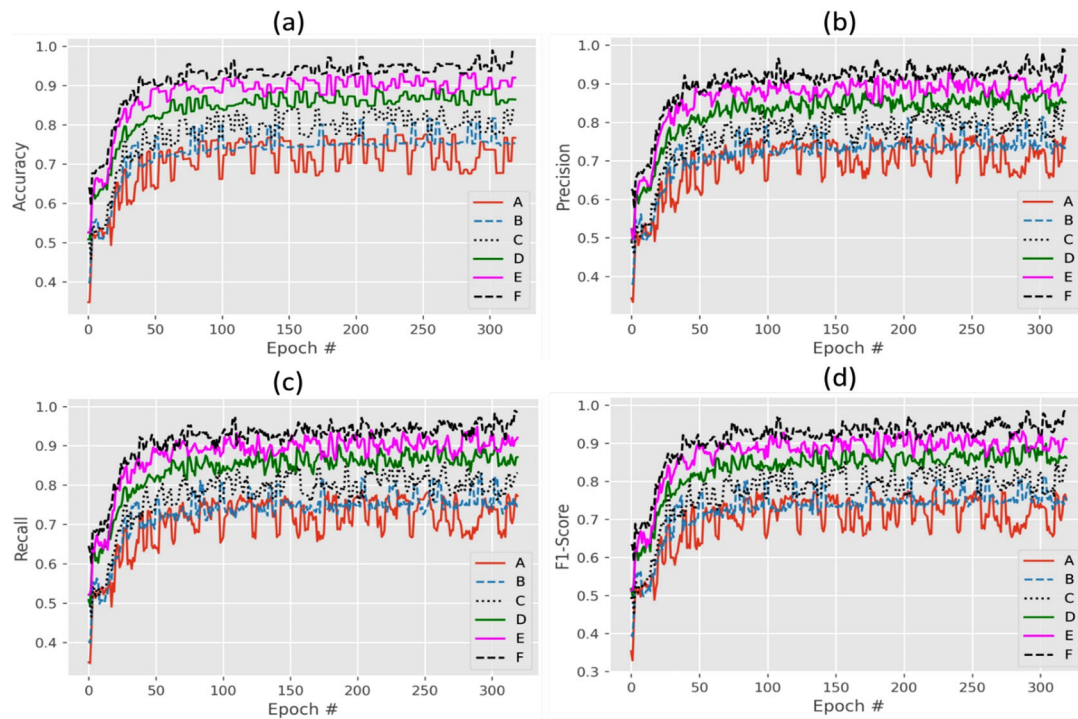
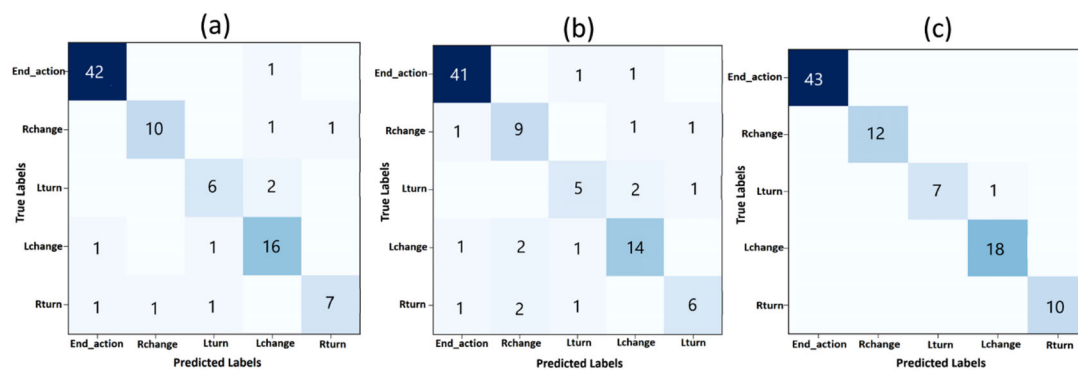
**FIGURE 8** The effect of using augmentation on (a) accuracy, (b) precision, (c) recall rate, and (d): F1-score. A = base, B = base + FlipLR, C = B + cutout, D = C + augmix, E = D + smooth + translate.**FIGURE 9** Confusion matrix for a) inside view b) outside view c) inside and outside views.

TABLE 8 Performance of the model using “inside view” and “outside view” dataset.

PH (s)	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
0 s	98.90	98.96	98.90	98.88
-1 s	93.40	93.58	93.41	93.39
-2 s	84.61	84.67	84.62	84.51
-3 s	71.42	70.24	71.43	70.22
-4 s	57.14	52.04	57.14	52.20

TABLE 9 Performance comparison of the proposed model with other state-of-the-art models on both “inside view” and “outside view” videos.

References	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)	PH (s)
Rong et al. [2]	83.98	N/A	N/A	84.3	0
Gebert et al. [20]	75.5	N/A	N/A	73.2	0
Ours	98.90	98.96	98.90	98.88	0

TABLE 10 Accuracy results of K -fold method using both “inside view” and “outside view” videos, $K = 5$.

PH (s)	Method	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Mean \pm std (%)
0 s	Our	96.70	97.80	97.80	96.70	98.90	97.58 \pm 0.92
0 s	Our + OTC	97.80	97.80	98.90	98.90	98.90	98.46\pm0.60
-1 s	Our	91.20	93.40	92.30	91.20	93.40	92.3 \pm 1.1
-1 s	Our + OTC	92.30	93.40	92.30	92.30	94.50	92.96\pm0.98
-2 s	Our	82.41	83.51	82.41	83.51	84.61	83.29 \pm 0.92
-2 s	Our + OTC	83.51	82.41	82.41	84.61	85.71	83.73\pm1.43
-3 s	Our	68.18	69.23	70.32	67.03	71.42	69.24 \pm 1.72
-3 s	Our + OTC	69.23	71.42	70.32	67.03	71.42	69.88\pm1.83
-4 s	Our	49.45	54.94	52.74	50.54	57.14	52.96 \pm 3.14
-4 s	Our + OTC	50.54	54.94	52.74	51.64	58.24	53.62\pm3.05

with the inside or outside view, only. Also, they indicate that the model’s error is not biased toward any of the specific classes but is instead distributed to all of the classes.

TABLE 11 Accuracy results of five classes of maneuvers under different prediction horizons using both “inside view” and “outside view” videos.

PH (s)	End action	Rchange	Lturn	Lchange	Rturn
0 s	100	100	87.5	100	100
-1 s	97.7	91.7	75.0	94.4	90.0
-2 s	95.3	83.3	62.5	77.8	70.0
-3 s	88.4	66.7	39.5	44.4	60.2
-4 s	60.1	35.0	32.5	35.1	45.5

4.6 | Discussion

Similar to any other research project, this study has some limitations. However, our main motivation was to address the limitations of existing approaches in terms of dealing with the spatiotemporal complexity of driver intention prediction. DIPNet’s design allows us to model the system dynamics smoothly and capture the complex interactions among features. While it is true that other machine learning algorithms, such as LSTM, can be applied to similar tasks, DIPNet has specific advantages over those methods, especially in the context of the task we focus on, where we are interested in an efficient light-weight model, with real-time prediction, and time prediction adaptability. There is no explicit or rigorous mathematical proof for deep learning based models, due to their black box feature extraction nature. However, our paper provides a diverse set of experimental results to validate the performance of our proposed method. We used objective KPIs such as accuracy, F1-score, recall, and precision to evaluate the effectiveness of our model for various scenarios. We also conducted extensive experiments to prove the efficiency of our approach and its ability to handle video sequences to predict driver intention. Also, We analyzed the contribution of different components of DIPNet to the overall performance. These analyses and visualizations provide strong evidence to support our claims for the effectiveness of DIPNet.

5 | CONCLUSION

In this study, a new deep neural network model is proposed to anticipate the driver maneuver intention a few seconds in advance. We examined the model in three different scenarios: considering the in-cabin context only (the driver), the out-cabin context only (the road), and both in and out contexts and fusing them to anticipate the driver’s intention or upcoming action.

In our proposed method, we used DenseNet121, LSTM, and the global attention module to extract features. Also, RAFT and FlowNet2 were employed to extract optical flow. In the first scenario, to avoid overfitting and enhance the proposed framework’s performance, different augmentation methods such as FlipLR, translate, cutout, and augmix, were served. By Utilizing the accuracy, precision, recall, and F1-score as our evaluation metrics, the proposed framework outperformed the state-of-the-art works for the “out-cabin” and “in/out cabin” datasets. As a possible future work aiming at further improvements, we may suggest the utilization of Swin transformer [45] as an efficient encoder to extract image features at a fast speed, which is an essential component in recognizing diver maneuvers in the real world.

AUTHOR CONTRIBUTIONS

Mahdi Bonyani: Conceptualization, formal analysis, investigation, methodology, software, visualization, writing - original draft. Mina Rahmani: Conceptualization, formal analysis, investigation, methodology, writing - original draft. Simindokht Jahangard: Conceptualization, data curation, investigation,

resources. Mahdi Rezaei: Methodology, resources, supervision, validation, visualization, writing - original draft, writing - review and editing.

ACKNOWLEDGEMENTS

As part of the Hi-Drive project, this research has received funding from the European Union's Horizon 2020 Research and Innovation Programme under grant No. 101006664. The article reflects only the authors' view and neither the European Commission nor the CIENA is responsible for any use that may be made of the information this document contains.

CONFLICT OF INTEREST

The authors declare no conflicts of interest.

DATA AVAILABILITY STATEMENT

The data that support the findings of this study are openly available in Brain4Cars repository at <https://www.brain4cars.com>. We have also made the DIPNet model publicly available to the research community at: <https://github.com/mbonyani/DIPNet>.

ORCID

Mahdi Rezaei  <https://orcid.org/0000-0003-3892-421X>

REFERENCES

- WHO, Road traffic injuries, World Health Organisation. <https://www.who.int/news-room/fact-sheets/detail/road-traffic-injuries> (2021). Accessed 21 June 2021
- Rong, Y., Akata, Z., Kasneci, E.: Driver intention anticipation based on in-cabin and driving scene monitoring. In: 2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC), pp. 1–8. IEEE, Piscataway, NJ (2020)
- Tonutti, M., Ruffaldi, E., Cattaneo, A., Avizzano, C.A.: Robust and subject-independent driving manoeuvre anticipation through domain-adversarial recurrent neural networks. *Rob. Auton. Syst.* 115, 162–173 (2019)
- Jain, A., Koppula, H.S., Soh, S., Raghavan, B., Singh, A., Saxena, A.: Brain4cars: car that knows before you do via sensory-fusion deep learning architecture. arXiv:1601.00740 (2016)
- Muhammad, K., Hussain, T., Ullah, H., Ser, J.D., Rezaei, M., Kumar, N., Hijji, M., Bellavista, P., de Albuquerque, V.H.C.: Vision-based semantic segmentation in scene understanding for autonomous driving: recent achievements, challenges, and outlooks. *IEEE Trans. Intell. Transp. Syst.* 23(12), 22694–22715 (2022)
- Gite, S., Agrawal, H., Kotecha, K.: Early anticipation of driver's maneuver in semiautonomous vehicles using deep learning. *Prog. Artif. Intell.* 8(3), 293–305 (2019)
- Jain, V., Liu, D., Baldi, S.: Adaptive strategies to platoon merging with vehicle engine uncertainty. *IFAC-PapersOnLine* 53(2), 15065–15070 (2020)
- Yurtsever, E., Lambert, J., Carballo, A., Takeda, K.: A survey of autonomous driving: common practices and emerging technologies. *IEEE Access* 8, 58443–58469 (2020)
- Ou, C., Karray, F.: Deep learning-based driving maneuver prediction system. *IEEE Trans. Veh. Technol.* 69(2), 1328–1340 (2019)
- Braunagel, C., Rosenstiel, W., Kasneci, E.: Ready for take-over? a new driver assistance system for an automated classification of driver take-over readiness. *IEEE Intell. Transp. Syst. Mag.* 9(4), 10–22 (2017)
- Braunagel, C., Kasneci, E., Stolzmann, W., Rosenstiel, W.: Driver-activity recognition in the context of conditionally autonomous driving. In: 2015 IEEE 18th International Conference on Intelligent Transportation Systems, pp. 1652–1657. IEEE, Piscataway, NJ (2015)
- Braunagel, C., Geisler, D., Rosenstiel, W., Kasneci, E.: Online recognition of driver-activity based on visual scanpath classification. *IEEE Intell. Transp. Syst. Mag.* 9(4), 23–36 (2017)
- Gite, S., Agrawal, H.: Early prediction of driver's action using deep neural networks. *Int. J. Inf. Retr. Res* 9(2), 11–27 (2019)
- Jain, A., Koppula, H.S., Raghavan, B., Soh, S., Saxena, A.: Car that knows before you do: anticipating maneuvers via learning temporal driving models. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 3182–3190. IEEE, Piscataway, NJ (2015)
- Zhou, D., Ma, H., Dong, Y.: Driving maneuvers prediction based on cognition-driven and data-driven method. In: 2018 IEEE Visual Communications and Image Processing (VCIP), pp. 1–4. IEEE, Piscataway, NJ (2018)
- Jain, A., Singh, A., Koppula, H.S., Soh, S., Saxena, A.: Recurrent neural networks for driver activity anticipation via sensory-fusion architecture. In: 2016 IEEE International Conference on Robotics and Automation (ICRA), pp. 3118–3125. IEEE, Piscataway, NJ (2016)
- Zhou, D., Liu, H., Ma, H., Wang, X., Zhang, X., Dong, Y.: Driving behavior prediction considering cognitive prior and driving context. *IEEE Trans. Intell. Transp. Syst.* 22(5), 2669–2678 (2020)
- Moussaid, A., Berrada, I., El Kamili, M., Fardousse, K.: Predicting driver lane change maneuvers using driver's face. In: 2019 International Conference on Wireless Networks and Mobile Communications (WINCOM), pp. 1–7. IEEE, Piscataway, NJ (2019)
- Rezaei, M., Azarmi, M., Mohammad Pour, F.: Traffic-Net: 3D traffic monitoring using a single camera. arXiv:2109.09165 (2021)
- Gebert, P., Roitberg, A., Haurilet, M., Stiefelwagen, R.: End-to-end prediction of driver intention using 3D convolutional neural networks. In: 2019 IEEE Intelligent Vehicles Symposium (IV), pp. 969–974. IEEE, Piscataway, NJ (2019)
- Olabi, O., Martinson, E., Chintalapudi, V., Guo, R.: Driver action prediction using deep (bidirectional) recurrent neural network. arXiv:1706.02257 (2017)
- Rekabdar, B., Mousas, C.: Dilated convolutional neural network for predicting driver's activity. In: 2018 21st International Conference on Intelligent Transportation Systems (ITSC), pp. 3245–3250. IEEE, Piscataway, NJ (2018)
- Rajaraman, S., Antani, S.K., Poostchi, M., Silamut, K., Hossain, M.A., Maude, R.J., Jaeger, S., Thoma, G.R.: Pre-trained convolutional neural networks as feature extractors toward improved malaria parasite detection in thin blood smear images. *PeerJ* 6, e4568 (2018)
- Dosovitskiy, A., Fischer, P., Ilg, E., Hausser, P., Hazirbas, C., Golkov, V., Van Der Smagt, P., Cremers, D., Brox, T.: Flownet: learning optical flow with convolutional networks. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2758–2766. IEEE, Piscataway, NJ (2015)
- Ilg, E., Mayer, N., Saikia, T., Keuper, M., Dosovitskiy, A., Brox, T.: Flownet 2.0: evolution of optical flow estimation with deep networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2462–2470. IEEE, Piscataway, NJ (2017)
- Rezaei, M., Klette, R.: Look at the driver, look at the road: no distraction! no accident! In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 129–136. IEEE, Piscataway, NJ (2014)
- Tonutti, M.: Domain adversarial RNN (DA-RNN) model. <https://zenodo.org/record/1009540> (2017). Accessed 12 October 2017
- Hou, Y., Rezaei, M., Romano, R.: Multi-level and multi-modal feature fusion for accurate 3D object detection in connected and automated vehicles. arXiv:2212.07560 (2022)
- Simonyan, K., Zisserman, A.: Two-stream convolutional networks for action recognition in videos. arXiv:1406.2199 (2014)
- Sharifi, A., Zibaei, A., Rezaei, M.: A deep learning based hazardous materials (HAZMAT) sign detection robot with restricted computational resources. *Mach. Learn. Appl.* 6, 100104 (2021)
- Teed, Z., Deng, J.: Raft: Recurrent all-pairs field transforms for optical flow. In: European Conference on Computer Vision, pp. 402–419. Springer, Cham (2020)

32. Zhai, M., Xiang, X., Lv, N., Kong, X.: Optical flow and scene flow estimation: A survey. *Pattern Recognit.* 114, 107861 (2021)
33. Ji, S., Xu, W., Yang, M., Yu, K.: 3D convolutional neural networks for human action recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* 35(1), 221–231 (2012)
34. Tran, D., Bourdev, L., Fergus, R., Torresani, L., Paluri, M.: Learning spatiotemporal features with 3D convolutional networks. In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 4489–4497. IEEE, Piscataway, NJ (2015)
35. Donahue, J., Anne Hendricks, L., Guadarrama, S., Rohrbach, M., Venugopalan, S., Saenko, K., Darrell, T.: Long-term recurrent convolutional networks for visual recognition and description. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2625–2634. IEEE, Piscataway, NJ (2015)
36. Yue-Hei Ng, J., Hausknecht, M., Vijayanarasimhan, S., Vinyals, O., Monga, R., Toderici, G.: Beyond short snippets: deep networks for video classification. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4694–4702. IEEE, Piscataway, NJ (2015)
37. DeVries, T., Taylor, G.W.: Improved regularization of convolutional neural networks with cutout. *arXiv:1708.04552* (2017)
38. Hendrycks, D., Mu, N., Cubuk, E.D., Zoph, B., Gilmer, J., Lakshminarayanan, B.: Augmix: a simple data processing method to improve robustness and uncertainty. *arXiv:1912.02781* (2019)
39. Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4700–4708. IEEE, Piscataway, NJ (2017)
40. Ghiasi, G., Lin, T.-Y., Le, Q.V.: Dropblock: a regularization method for convolutional networks. *arXiv:1810.12890* (2018)
41. Hrnjica, B., Bonacci, O.: Lake level prediction using feed forward and recurrent neural networks. *Water Resour. Manage.* 33(7), 2471–2484 (2019)
42. Luong, M.-T., Pham, H., Manning, C.D.: Effective approaches to attention-based neural machine translation. *arXiv:1508.04025* (2015)
43. Wang, Y., Huang, G., Song, S., Pan, X., Xia, Y., Wu, C.: Regularizing deep networks with semantic data augmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* 44(7), 3733–3748 (2021)
44. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. *arXiv:1412.6980* (2014)
45. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: hierarchical vision transformer using shifted windows. *arXiv:2103.14030* (2021)

How to cite this article: Bonyani, M., Rahmanian, M., Jahangard, S., Rezaei, M.: DIPNet: Driver intention prediction for a safe takeover transition in autonomous vehicles. *IET Intell. Transp. Syst.* 17, 1769–1783 (2023). <https://doi.org/10.1049/itr2.12370>