

Context-driven Multi-stream LSTM (M-LSTM) for Recognizing Fine-Grained Activity of Drivers

Ardhendu Behera^[0000-0003-0276-9000], Alexander Keidel^[0000-0003-2804-602X],
and Bappaditya Debnath^[0000-0002-2026-8632]

Department of Computer Science, Edge Hill University, Ormskirk, L39 4QP, UK
<https://www.edgehill.ac.uk/computerscience/>
{beheraa, keidela, debnathb}@edgehill.ac.uk

Abstract. Automatic recognition of in-vehicle activities has significant impact on the next generation intelligent vehicles. In this paper, we present a novel Multi-stream Long Short-Term Memory (M-LSTM) network for recognizing driver activities. We bring together ideas from recent works on LSTMs, transfer learning for object detection and body pose by exploring the use of deep convolutional neural networks (CNN). Recent work has also shown that representations such as hand-object interactions are important cues in characterizing human activities. The proposed M-LSTM integrates these ideas under one framework, where two streams focus on appearance information with two different levels of abstractions. The other two streams analyze the contextual information involving configuration of body parts and body-object interactions. The proposed contextual descriptor is built to be semantically rich and meaningful, and even when coupled with appearance features it is turned out to be highly discriminating. We validate this on two challenging datasets consisting driver activities.

1 Introduction

Recognition and description of human action/activities in videos and images is a fundamental challenge in computer vision. Over the last two decades, it has been extensively studied and has generated a rich volume of literature [15, 2]. It has received increasing attention due to far-reaching applications such as intelligent video surveillance, robotics and AI, human computer interactions, sports analysis, autonomous and intelligent vehicles. Recognising videos requires analysing spatio-temporal data, as well as effective processing and representation of visual and temporal information. Over the years, this representation is dominated by hand-crafted features such as space-time interest points [23, 22], joint shape and motion descriptors [39, 4, 24], feature-level relationships [32, 21] and object-hand interactions [9, 13, 3], due to their superior performance. This has been challenged by the recent advances in Deep Convolutional Neural Network (DCNN) [11, 26, 12]. However, extending these networks on video analysis (i.e temporal data) introduce many new challenges, which are often addressed using temporal modeling. Recently, Long Short-Term Memory (LSTM), a specialized form of

Recurrent Neural Network (RNN) is often used to handle temporal data [18]. This is mainly due to the fact that it can encode state, capture temporal ordering and long range dependencies. LSTMs combined with CNNs have shown great performance in video classification tasks [8, 27], learning long-term motion dependencies and spatial-temporal relations [25] and precipitation nowcasting [42]. However, it increases network complexity that requires training of a very large number of parameters and tuning many different hyper-parameters. This could be challenging, especially in real-world applications (e.g. robotics and autonomous vehicles) in which there are constraints on power, processing time, size, area and weight.

Recently, there is a growing interest to address the above-mentioned problem via *transfer learning* (TL), aiming to reduce training time and improve performance [43, 29]. The initial convolutional layers in deep CNNs produce features with a surprising level of generality (i.e. useful for most images) [43, 29]. This generality is a key characteristic of TL that influences the initialization of a target network with layers and trained weights from a base network and is very effective. However, for video-based human activity recognition, most works focus on image-based TL but less work has been done on video-based TL and the best way to do this is still an open question.

In the context of intelligent and (semi-)autonomous vehicles, there is a prominent role of understanding and predicting in-vehicle activities. This would also allow monitoring driver activity (e.g. use of phone, eating and drinking, etc.) and readiness for a takeover request (TOR) [20] in AVs, defined by the National Highway Traffic Safety Administration (NHTSA). This is also a step toward the eventual implementation of the “cognitive car” [14] and self-learning autonomous vehicles (AVs) [6] concepts, which are aimed to learn from the in-vehicle activities to provide a better experience for its occupants and optimize their performance. In this work, we focus on fine-grained in-vehicle (e.g. driver) activity recognition. The term *fine-grained* is similar to the one in [30], aims to distinguish between activities involving little differences. The drivers’ activity can be seen as a fine-grained recognition problem (e.g. texting vs talking over phone).

In this paper, we propose a novel deep neural network called Multi-stream LSTM (M-LSTM) for recognising fine-grained activities. The proposed network benefits from the TL by using per-frame CNN features from different layers of available pre-trained CNN models (e.g. VGG16 [34]) as appearance features. We evaluate our M-LSTM network from one stream upto four streams. Our network is flexible and if required, it can accommodate more input streams depending on the target application. The goal is to maximize the use of TL in order to minimize the training complexity and resources while still achieving competitive performance on this fine-grained activity recognition task. This work includes the following novel contributions:

- We demonstrate the effectiveness of our novel Multi-stream LSTM (M-LSTM) for fine-grained activity recognition task. The network is light-weight and can be trained using CPU. It is flexible to accommodate more streams.

- We explore the benefit of TL and validate the significance of context represented by high-level knowledge involving our novel body pose and body-object interactions descriptor. The inclusion of context leads to significant improvements in results. Although LSTMs have been used for action recognition, but in this work we analyze the importance of contextual information influences the way LSTMs are used.
- We are the first to report the video-based activity recognition using the State Farm dataset [7] and the “Distracted Driver” dataset [1], which are aimed to recognise driver’s state/activity. All the existing approaches [16, 1, 36] are based on the single image classification.

2 Related Works

Video-based human activity recognition has made considerable progress. Traditional approaches described in [15, 2] are based on hand-crafted features. Recently, these hand-crafted features are replaced with the deep features due to their superior performance. Wang *et al* [40] replaced the hand-crafted features with CNN features and stacked optical flow, resulting in improved performance. Simonyan and Zisserman [34] have used a two-streams network for action recognition in which video frames and stacked optical flow are fed as two separate streams. In [33], Ryoo *et al* used pooled feature representation, which gave superior performance using CNN features.

Long Short-Term Memory (LSTM) models have shown great performance in activity recognition and often used to combine multiple streams of information [8, 35, 44]. Singh *et al* [35] have shown that combining full image features with bounding box features improves performance in video classification for fine-grained actions. Wu *et al* [41] combine several streams: a spatial CNN fed into an LSTM, an optical flow CNN fed into a second LSTM, and an audio spectrogram CNN for video classification. LSTMs have also shown improved performance over two-streams CNNs in recognising activities [8, 44].

The traditional vision-based in-vehicle activity monitoring approaches are mostly focused on cues involving driver’s upper-body parts (e.g. face, eye, hand and head) and their movements [19, 28, 38]. These approaches are often targeted at automatic detection of safe/unsafe driving behaviors (e.g. drowsiness, fatigue, distractions, emotions, etc.) using hand-crafted features (e.g. LBP, HOG, Haar-like) combined with classical machine learning algorithms such as SVM and AdaBoost. Understanding driver’s activities (e.g. using phone, eating, drinking, etc.) is vital not only for safe driving but also for the autopilot hand-over process for the next generation self-learning AVs. Recently, there has been some progress in using CNN models in monitoring [1, 16, 36]. However, the adaptation of the state-of-art CNN models driven by the contextual information is yet to be explored. In this paper, we aim to address this.

A good progress has been made in recognising activity using latest approaches such as LSTMs and CNNs. Most of these models are trained on very large datasets and often requires multiple days, even when GPUs are used. It has

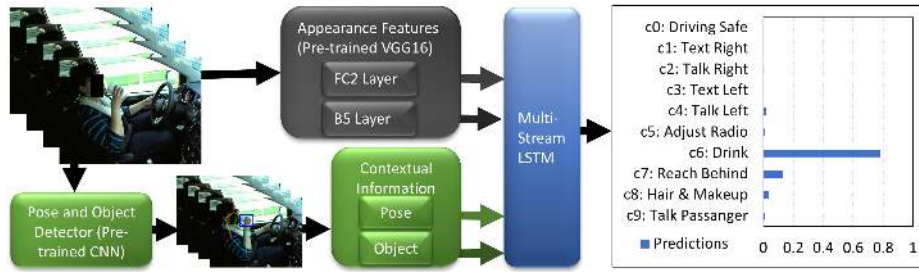


Fig. 1: Overview of the proposed Multi-stream LSTM (M-LSTM) for driver’s activity recognition

also been shown that the action recognition performance in still images has significantly improved by incorporating person-objects interactions [11, 26] and contextual cues such as body pose [12]. These cues are also vital for video-based activity recognition. Such cues are affiliated to pixel-level and therefore, incorporating these cues in existing LSTMs and CNNs would result in further increase in complexity of these models for video-based activity recognition. As a result, it would be difficult to adapt these models in applications targeted to robotics and autonomous systems. In this work, we revise these contextual cues and represent it as a high-level contextual knowledge that encodes body-pose and hand-object interactions by considering pairwise relationships. These relationships are extracted by exploring the per-frame configuration of the body parts and objects. This is feasible due to the recent development of the state-of-the-art objects [17] and body-parts [5] detector to operate in real-time. We also explore the suggestion in [29] to extract static appearance feature using TL via deep image classification network such as VGG16 [34]. Here we make the observation that use of different level of abstractions (i.e. from different layers) is very useful. We propose a novel Multi-stream LSTM (M-LSTM) which is relatively shallow (upto 8 layers) to integrate contextual cues, long-term sequence information and different levels appearance feature to recognize fine-grained activity of a driver.

3 Proposed Activity Recognition Approach

The overview of the proposed framework is shown in Fig.1. The architecture has three main components: 1) Transferable deep CNN features, 2) contextual cues involving body pose and body-object interaction and 3) the proposed Multi-stream LSTM (M-LSTM) for sequence modeling and activity recognition.

3.1 Transferable deep CNN Features

Most of the state-of-the-art deep CNN pre-trained models are publicly available. These models are trained on a large dataset such as ImageNet [31]. Such models learn from very general (e.g. Gabor filters, edges, color blobs) to task-specific



Fig. 2: Contextual descriptors capturing body pose and body-objects interactions: a) detected body joints and cup as an object of interest, b) pairwise relations between nose and the rest of the body joints, c) all possible pairwise relations between detected body joints, and d) pairwise relations between detected cup and various body joints.

features as we move from first-layer to the last-layer [43] and thus, often applied to new dataset with no/minimal fine-tuning. Therefore, it allows us to leverage their power for video analysis when using them as feature extractors. We use VGG16 [34] to extract features at two different extraction points: 1) Block5 (B5) pooling and 2) FC2 (Fully connected). The aim is to extract appearance features denoting various level of abstraction to compare their suitability for a given task.

3.2 Contextual Descriptors

In this work, context refers to the representation of high-level knowledge involving human pose and human-object interactions. Our contextual descriptors are aimed to represent this knowledge effectively. Human action is often perceived from the body pose i.e. configuration of body parts in images. This configuration often provides discriminative appearance cues in differentiating various actions (e.g. standing vs sitting vs bending). However, many fine-grained non-driving activities (e.g. texting, talking over phone, eating, drinking, etc.) exhibit similar body parts configuration. Thus, it is difficult to distinguish them using only body parts. In such cases, involved objects (e.g. cup, bottle, phone, etc.) and its interaction with the body parts play a key role in differentiating these activities. Therefore, we use contextual descriptors to represent relationships between body parts and objects, as well as between various body parts (Fig. 2).

Body pose descriptor The proposed body pose descriptor translates the body parts configuration to a feature vector by encoding relationships between various body parts (Fig. 2c). We use the state-of-the-art Part Affinity Fields (PAFs) [5], which can detect the body parts of multiple person in real-time. It gives output as location (i.e. x, y position in image plane) of 18 body joints: 1) nose, 2) neck, 3) right shoulder, 4) right elbow, 5) right wrist, 6) left shoulder, 7) left elbow, 8) left wrist, 9) right hip, 10) right knee, 11) right ankle, 12) left hip, 13) left knee, 14) left ankle, 15) right eye, 16) left eye, 17) right ear and 18) left ear. We use the

upper-body (knee and above) and therefore, 16 joints (except both ankles) are considered. There are inevitable noises (missing joints and false detection) and is mainly due to occlusions and contents resulting from driving circumstances and environmental situations. Therefore, detecting all joints accurately would be difficult even if one fine-tuned/re-trained the model on the target dataset. Our goal is to minimize this noise while creating the descriptor and thus, we consider pairwise relations between all possible detected joints. For example, if an elbow is noisy (false detection or undetected) then the relationships between other detected joints (e.g. neck, shoulder, wrist, etc.) would be able to capture the body pose.

There are 16 joints, resulting 120 ($\frac{16 \times 15}{2}$) possible unique pairs. For each pair, we compute a relational feature f . Let's consider a pair of joints j_1 and j_2 , located at (x_1, y_1) and (x_2, y_2) , respectively. Their relationship is represented using distance $r = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$ and orientation $\theta = \arctan(\frac{y_2 - y_1}{x_2 - x_1})$. The angle θ is binned into h number of bins and the magnitude r contributes to the respective bin(s) where the θ falls into. As a result, f is sparse and its dimension is the number of bins h . We apply the L_2 normalisation to f . The process continues for all 120 pairs and concatenate them to represent our pose descriptor $D_p = [f_1, f_2, \dots, f_{120}]$ of length $120 \times h$.

Body-object descriptor Similar to the pose descriptor, our body-object descriptor captures the pairwise relationship between the body joints and involved objects. This relationship encodes the relative position of an object with respect to a given joint in a scene. Thus, we need to detect the commonly used objects (e.g. mobile phone, water bottle, cup, etc.). Similar to the body joints, we use the TL approach for objects detection i.e. using a pre-trained detector on the target dataset. We benefit from the state-of-the-art deep CNN models, which have achieved remarkable results. One such model is the combination of Faster R-CNN with Inception ResNet-V2 [17]. This model is trained on COCO dataset consisting 330K images, 1.5 million objects instances and 80 object categories.

Our focus is on the *object of interest* (e.g. phone, bottle, cup, etc.). A common observation is that the size of these objects is small with respect to the size of the driver (Fig. 2d) and appears in the vicinity of the driver's bounding box. Thus, we use the bounding box information (size and aspect ratio) to select the objects of interest. We could have selected these based on their types. However, we noticed that there are noises (e.g. wrongly labeled) in detection and is mainly due to occlusion by the driver's hand, as well as the use of TL since the detector is trained on a different dataset. It is observed that often mobile phones and coffee mugs are detected as a remote, cup as a wine glass. Our aim is to model contextual cues (configuration of objects with respect to joints) to discriminate the fine-grained activities. Thus, we argue that if an object is wrongly labeled, the combined configuration of body joints and object would provide enough cues for discriminating various activities. For example, if a phone is labeled as a mug then based on the arm configuration, its position with respect to other

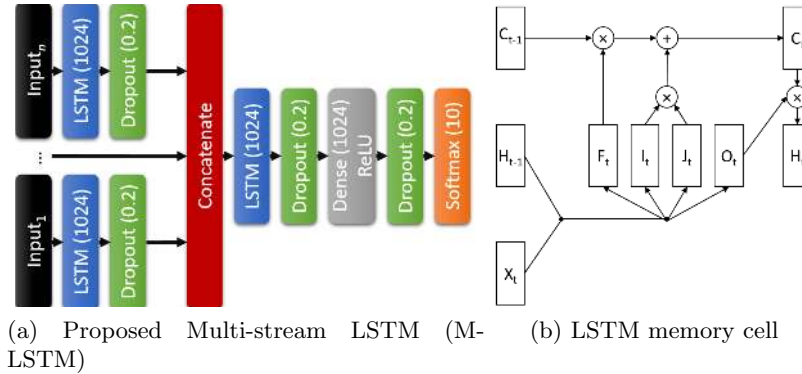


Fig. 3: a) Proposed relatively shallow (up to 8 layers) Multi-stream LSTM (M-LSTM) and b) an LSTM memory cell used in this work from [18].

body parts (e.g. torso, head, etc.) and the object’s position with respect to body parts, would provide cue in discriminating *texting* vs *talking* vs *drinking*.

A total of 25 objects of interest are selected on the target datasets [7, 1] by considering their relative size (area $< 1/4$ th of the driver’s bounding box) and position (bounding box overlap $> 80\%$) with respect to the driver’s bounding box. The proposed body-object descriptor (D_o) captures the pairwise relationship between 16 body joints and the detected objects. D_o encodes this relationship as a histogram of oriented relation \hat{f} (Fig. 2d), which is computed similar to the body joints relational feature f (Fig. 2b) in the pose descriptor D_p . A total of 400 (25×16) pairwise relations ($\hat{f}_1 \dots \hat{f}_{400}$) are stacked to represent our body-object descriptor $D_o = [\hat{f}_1, \hat{f}_2, \dots, \hat{f}_{400}]$ of length $400 \times h$ (h angle bins).

3.3 Multi-stream Long Short-Term Memory (M-LSTM) Network

The proposed Multi-stream LSTM (M-LSTM) network for fine-grained activity recognition is shown in Fig. 3a. The aim is to combine multiple feature types in order to take the best advantage of data representation with multiple levels of abstractions and allow the model to learn activities from these representations. The proposed M-LSTM is inspired by [10]. It is light-weight and consists of LSTM, Dropout, FC and Softmax layers. The architecture is flexible so that more input streams could easily be added and takes advantage of off-the-shelf CNN features, which have shown impressive performance in visual recognition tasks [29, 43]. The inputs consist of per-frame appearance and contextual features. The sequential information in the M-LSTM is captured by the two LSTM layers - one is in individual stream and the other is after the fusion (Fig. 3a). It should be noted that all our input features are based on transfer learning i.e CNN features, object and body parts detectors are not trained/fune-tuned on the target dataset.

LSTM is a special type of Recurrent Neural Network (RNN). It is capable of learning long-term dependencies by incorporating memory units that allow the network to learn, forget previous hidden states and update hidden states, when required [8]. The M-LSTM network uses the LSTM architecture described in [18] (Fig. 3b). At a given timestep t , the M-LSTM takes input $x_t = [D_p^t, D_o^t, F_1^t, F_2^t]$, consisting body pose D_p^t , body-object interactions D_o^t , and CNN feature F_1^t and F_2^t extracted from the respective FC2 and Block5 layer of the VGG16 [34]. The model updates at time t given the memory cells for long-term c_{t-1} and short-term h_{t-1} recall from the previous timestep $t - 1$ and is by:

$$\begin{aligned}
 i_t &= \tanh(W_{xi}x_t + W_{hi}h_{t-1} + b_i) \\
 j_t &= \text{sigm}(W_{xj}x_t + W_{hj}h_{t-1} + b_j) \\
 f_t &= \text{sigm}(W_{xf}x_t + W_{hf}h_{t-1} + b_f) \\
 o_t &= \tanh(W_{xo}x_t + W_{ho}h_{t-1} + b_o) \\
 c_t &= c_{t-1} \odot f_t + i_t \odot j_t \\
 h_t &= \tanh(c_t) \odot o_t
 \end{aligned} \tag{1}$$

Where W_* denotes weight matrices, b_* biases, \odot element-wise vector product, respectively. The LSTM has two kinds of hidden states: c_t and h_t which allow it to make complex decisions over a short period of time. It also includes an input gate i_t , input modulation gate j_t contributing to memory, forget gate f_t , output gate o_t as a multiplier between the memory gates (Fig. 3b). The gates i_t and f_t can be seen as knobs allowing the LSTM to selectively consider its current input or forgets its previous memory. Similarly, the output gate o_t learns how much memory cell c_t need to be transferred to the hidden state h_t . These additional memory cells give the ability to learn extremely complex and long-term temporal dynamics in comparison to the RNN. Moreover, LSTM provides the ability to remember information and recall it at a later point in time when needed and is suitable for solving video recognition problems.

4 Experiments, Results and Discussion

We use the State Farm [7] and ‘‘Distracted Driver’’ [1] dataset, comprised of inwards facing dashboard camera images depicting ten fine-grained activities: c0) safe driving, c1) texting - right, c2) talking on the phone - right, c3) texting - left, c4) talking on the phone - left, c5) operating the radio, c6) drinking, c7) reaching behind, c8) hair and makeup, and c9) talking to passenger.

In the State Farm [7] dataset, there are 260 clips (22,424 images) from 26 drivers (mixture of male and female from different ethnicity). There are 10 clips (one for each activity) per driver. Similarly, in the ‘‘Distracted Driver’’ dataset [1], there are 310 clips (17,308 images) from 31 drivers. This dataset has also 10 classes and 10 clips per driver. In our experiment, we uniformly sampled 30 frames per clip. Let’s say there are m number of frames in a given clip and we wish to sample the desired $n = 30$ frames by selecting a frame at position $j = 0 \dots m - 1$ in the original clip, where $j = \lfloor \frac{i \times m}{n} \rfloor$ for $i = 0 \dots n - 1$. For our experiment, 70% (180 videos from 18 drivers in [7] and 220 videos from 22

drivers in [1]) of the dataset is used for training and the rest 30% (80 and 90 clips form the rest of the 8 and 9 drivers in [7] and [1], respectively) for validation. We select this split so that the validation set consists of entirely unseen drivers.

We explore all possible permutations using 4 different features. Our experiments provide multiple outcomes: 1) performance of transferable features from different layers (B5 pooling and FC2) of VGG16 [34], 2) performance of contextual descriptors like body pose and body-object interaction in comparison to CNN features, 3) the impact on performance using various combinations of features, and 4) the influence of temporal information (number of frames) on performance for live monitoring.

We use default image size (224×224) for CNN features (B5 and FC2) using pre-trained VGG16 [34], resulting feature length of 4096 (FC2) and 25088 (B5). For our contextual descriptors, we have experimented with different number of bins ($h = 6, 9, 12$ and 18) and found better performance for $h = 12$, resulting the size 1440 (120×12) and 4800 (400×12) for the pose and body-object interaction descriptor, respectively.

In the proposed M-LSTM, the number of layers, their orders and parameters are selected based on the performance. The final M-LSTM is shown in Fig. 3a. Each clip in the dataset consists of a single activity and therefore, we are interested in the class probability distribution once M-LSTM has observed the entire sequence. To achieve this, many approaches exist [44]: (1) using the prediction at the last frame of a given clip; (2) max pooling the predictions over the entire clip; (3) summing all of the frames predictions over time and returning the most frequent. We have experimented our model by using approaches (1) and (2). Using approach (1) i.e. without temporal pooling layer, we have observed the per-frame accuracy, its effect on the number of input frames and the minimum number of frames required for a good early prediction. The models are trained using the RMSprop [37] optimizer to minimize the categorical cross entropy $L_v = -\sum_c y_{v,c} \log(p_{v,c})$, where p are the predictions, y are the targets, v denotes the training video and c denotes the class. One-stream model is trained using a learning rate (lr) of 2×10^{-5} ; two-, three- and four-streams of 5×10^{-5} , with all other parameters are assigned with default values. A Linux PC (Intel i7-5930K, 12 cores, 3.5 GHZ) with NVIDIA Quadro P6000 24GB GPU is used for our experiments. The models are trained for 50 epochs with a batch size of 32. Training time of each model is just under 20 minutes. The same training takes around 2:37h using CPU, which is still a viable option.

For evaluation, we use accuracy (ACC) and average precision (AP). ACC assigns equal cost to false positives and false negatives. Whereas, AP summarizes precision-recall curve. We also compute multi-class log loss $\logLoss = -\frac{1}{V} \sum_v \sum_c y_{v,c} \log(p_{v,c})$, where v represents test videos, c denotes activity labels, p implies predictions and y denotes targets. It quantifies the accuracy of a classifier by penalizing confident false classifications. For example, if a classifier assigns a very small probability to a correct class then the corresponding contribution to the log loss will be very large. An ideal classifier will have zero log loss. The performance of the M-LSTM is shown in Table 1. There are four sets

Table 1: Performance of the proposed M-LSTM: from one-stream to four-streams using State Farm [7] (left column) and “Distracted Driver” [1]. The performance is the *argmax* of the output from the softmax layer. All values are in percentages except for the log loss. Lower value of the log loss is better. The best performance is shown in bold for a given dataset with one or more input streams

	ACC	AP	Log Loss	ACC	AP	Log Loss
	State Farm [7] dataset			Distracted Driver [1] dataset		
One-stream						
Pose	48.75	60.00	5.61	12.22	12.20	2.25
FC2	52.50	72.50	2.70	30.00	34.18	2.44
Object	61.25	75.25	3.06	42.22	44.23	2.40
B5	77.50	85.00	1.32	38.89	47.46	1.94
Two-streams						
FC2+Pose	60.00	77.50	2.23	34.44	38.49	2.38
Pose+Object	61.25	73.75	5.01	44.44	48.64	4.00
B5+Pose	81.25	91.25	1.05	34.44	43.51	2.02
FC2+B5	81.25	88.75	0.99	36.67	49.23	1.88
FC2+Object	77.50	81.25	1.10	41.11	54.69	1.87
B5+Object	85.00	90.00	0.69	43.33	53.16	1.97
Three-streams						
FC2+Pose+Object	76.25	88.75	1.68	47.78	57.68	2.08
FC2+B5+Pose	78.75	87.50	1.16	34.44	40.22	2.28
FC2+B5+Object	86.25	96.25	0.51	46.67	52.83	1.75
B5+Pose+Object	87.50	96.25	0.62	52.22	59.66	1.66
Four-streams						
FC2+B5+Pose+Object	91.25	95.00	0.45	37.78	53.11	1.72

of rows representing the performance of one-stream to the four-streams. The left column is for the State Farm dataset [7] and the right column is for the “Distracted Driver” [1] dataset. The given performance is based on our proposed M-LSTM without temporal pooling in Fig. 3a. The performance is measured as the *argmax* of the final softmax layer. The best performance is shown as bold within a given set. It is clear that as we add more streams the performance improves in both the datasets.

Performance on State Farm dataset [7] For CNN features (FC2 and B5), the ACC of B5 is 25% better than the FC2 (Table 1, left column). In [29], CNN feature from FC layer is used for the visual recognition task. This shows the CNN features are dependent on the target dataset type and more than one extraction point should be considered while using transfer learning. Moreover, when we combine features from multiple extraction points (FC2+B5), the performance is better than the single ones. When our contextual information (body pose and body-object interactions) is added, the ACC increase by 10% (B5+Pose+Object) in comparison to the B5 alone. Similarly, adding this information to FC2, the performance increased by 33.75% and this explains the significance of our high-level

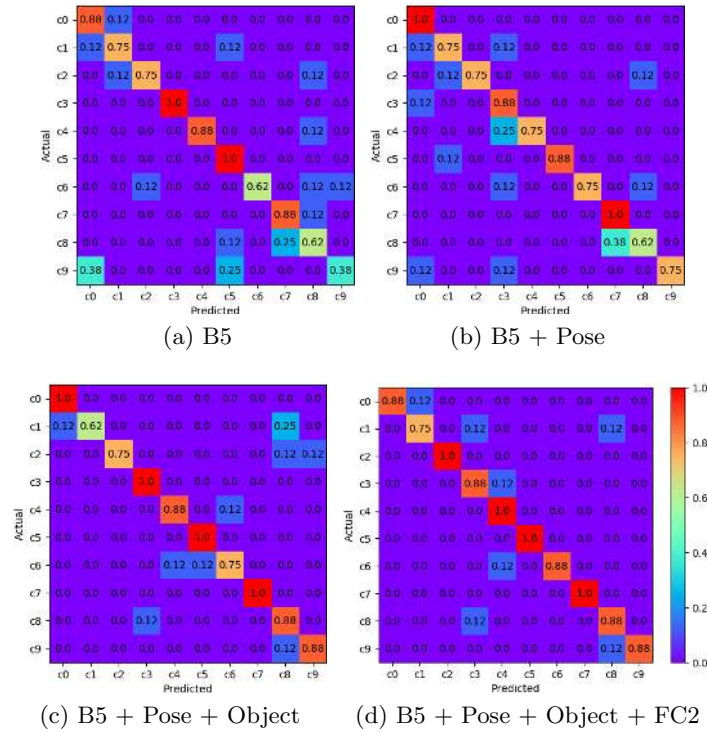


Fig. 4: Confusion matrices for one-stream to four-streams based on B5.

contextual descriptors. Our model gives the best performance (ACC: 91.25%), when all four-streams are used. The confusion matrix for one-stream to four-streams using the B5, is shown in Fig. 4. If we compare the one-stream (Fig. 4a) with four-streams (Fig. 4d), the performance of most of the activities is improved or same except the activity c0 - *safe driving* and c1 - *texting right*. The c0 is confused with c1. This could be due to both c0 (both hand on steering) and c1 exhibit similar body pose and the cell phone is often occluded because of the dashboard camera position. The four-streams performance of activity c3 - *texting - left* drops by 12% in comparison to the three-streams model (Fig. 4d vs 4c). This 12% is confused with the c4 - *talking on the phone - left*. This is mainly due to one of the subject’s left hand is close to the head while texting and based on the subject’s pose and phone position with respect to the body, the model recognized as talking left and could be the influence of contextual information. Whereas, using B5, the model recognises this one correctly (Fig. 4a).

Performance on Distracted Driver dataset [1] The performance of the proposed approach using the “Distracted Driver” dataset [1] is presented in Table 1 (right column). Similar to the State Farm [7], the performance increases as we

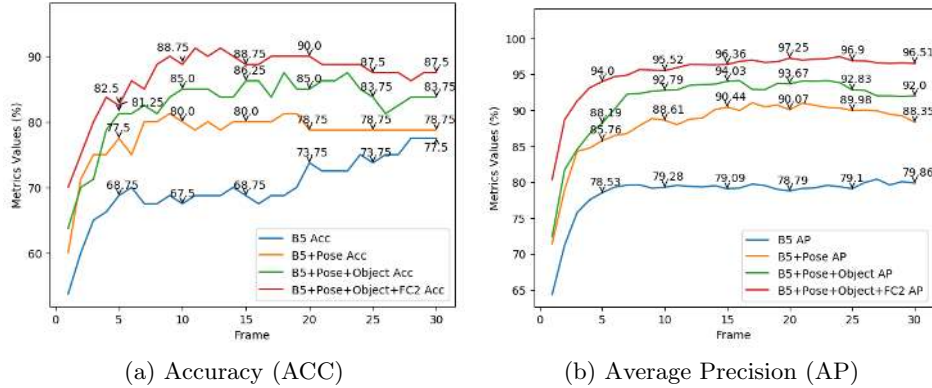


Fig. 5: Activity recognition performance (average over 8 drivers and each with 10 activity classes) for one-stream to four-streams based on B5. The proposed M-LSTM model’s ACC (a) and AP (b) over the model’s memory duration in frames using the State Farm dataset [7].

add more streams. However, the overall performance is quite low in comparison to the State Farm [7]. This could be due to the fact that the data in [1] was being collected from seven different countries in four different cars with several variations in driving conditions. Whereas, State Farm [7] data was collected using one car in a controlled environment i.e. a truck dragging the car around on the streets - so the drivers weren’t really driving.

In one-stream, the standout performance (ACC: 42.22%) is our contextual descriptor using body-object interactions (Table 1, right column). When this descriptor is combined with others, the overall performance is improved (B5+Object: 43.33%, Pose+Object: 44.44% and FC2+object: 41.11%). This demonstrates the significance of our proposed context-driven model. The best performance on this dataset is the combination of three streams i.e. B5+Pose+Object (ACC: 52.22%). When the fourth stream FC2 is integrated to it, the performance dropped to 37.78%. Therefore, the FC2 feature is not as good as the B5. A similar trend was observed in the State Farm [7] dataset as well.

Performance using temporal pooling layer We have experimented with the use of temporal pooling (max pooling) layer. This temporal pooling layer is added after the last LSTM layer (Fig. 3a), replacing the dropout layer. The performance is presented in Table 2 for the State Farm [7] dataset. Most of the time, the M-LSTM performs better without the temporal pooling (Table 1). The other notable observation is the log loss using max pooling. The performance is better (lower is better) than without the max pooling, except in the four-streams model. This implies the M-LSTM is more confident (high probability) in making right decision, when max pooling layer is used.

Table 2: Performance of the M-LSTM: from one-stream to four-streams with temporal pooling (*max pooling*), evaluated on the State Farm [7] dataset. All the values are in percentage except for the log loss. The best performance is shown in bold for a given combination(s) of input stream(s)

M-LSTM with max pooling using State Farm [7] dataset							
	ACC	AP	Loss		ACC	AP	Loss
One-stream				Two-streams			
Pose	35.00	55.00	2.23	FC2+Pose	62.50	82.50	1.31
FC2	51.25	71.25	1.96	Pose+Object	56.25	72.50	1.72
Object	55.00	72.50	1.64	B5+Pose	75.00	80.00	0.85
B5	60.00	80.00	1.19	FC2+B5	75.00	87.50	0.88
				FC2+Object	78.75	86.25	0.92
				B5+Object	81.25	92.50	0.53
Three-streams				Four-streams			
FC2+Pose+Obj	76.25	88.75	0.91	FC2+B5+	87.50	95.00	0.50
FC2+B5+Pose	75.00	87.50	0.76	Pose+Object			
FC2+B5+Obj	83.75	95.00	0.49				
B5+Pose+Obj	86.25	91.25	0.52				

M-LSTM memory duration We have also looked into the M-LSTM memory duration with respect to the number of frames. The ACC and AP for one-stream to four-streams using the State Farm dataset [7] with memory of 1 to 30 frames is shown in the Fig. 5a and Fig. 5b, respectively. The ACC using B5 (Fig. 5a) increases with the number of frames (still upward at frame 30). This means the M-LSTM needs more evidence for making the correct decision. Whereas, with contextual information (B5+pose+object), the accuracy reaches close to the maximum around frame 10. This shows our contextual information is semantically rich and meaningful to recognise activities with partial information (less number of frames). Therefore, the proposed M-LSTM could be used for live monitoring of the activities from partial observations. The accuracy of individual class using all four streams is shown in Fig. 6.

Performance comparison with state-of-the-art As mentioned earlier, we are the first to report video-based activity recognition on these datasets [7, 1]. The existing approaches [16, 1, 36] were evaluated using still images. For still images, Hssayeni *et al* [16] reported the accuracy of 85% using State Farm [7] and Abouelnaga *et al* [1] has achieved accuracy of 95.17% using their “Distracted Driver” dataset. However, [1] has used the validation images from the seen drivers i.e. for a given driver and activity, part of the video frames used in training and the rest for testing. In our experiments, we use entirely unseen drivers for testing.

5 Conclusion

We have developed a Multi-stream LSTM (M-LSTM) network for recognizing fine-grained activities of drivers. The network is light-weight and flexible to ac-

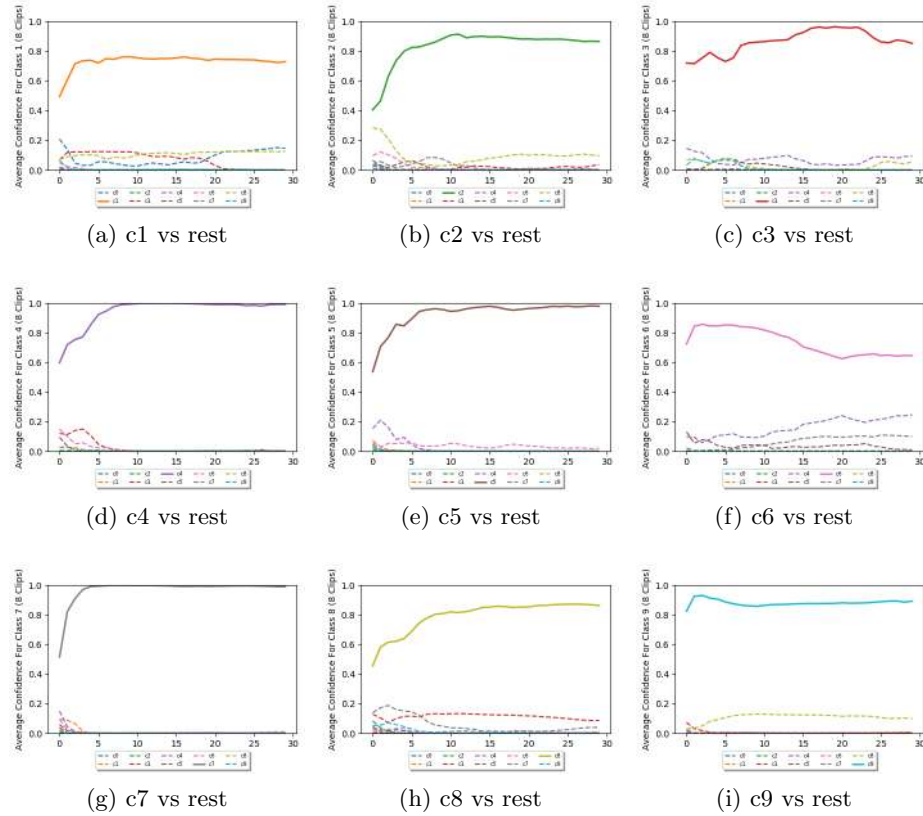


Fig. 6: Individual activity performance (average accuracy over 8 drivers) of our M-LSTM over the model’s memory duration in frames. The accuracy using four-streams (B5+Pose+Object+FC2): the solid line is the predicted activity and the dotted lines are the rest of the activities using the State Farm dataset [7].

commodate one or more input streams. We demonstrated how our proposed network learns to recognize fine-grained activities by exploring transfer learning and combining features with different levels of abstractions, as well as contextual features involving body pose and body-objects interactions. We further analyzes the suitability of the M-LSTM for activity recognition from partial observation. We believe this will help advance the field of in-vehicle activity recognition.

Acknowledgments: The research is supported by the Edge Hill University’s Research Investment Fund (RIF). We would like to thank Taylor Smith in State Farm Corporation for providing information about their dataset. The GPU used in this research is generously donated by the NVIDIA Corporation.

References

1. Abouelnaga, Y., Eraqi, H.M., Moustafa, M.N.: Real-time distracted driver posture classification. arXiv preprint arXiv:1706.09498 (2017)
2. Aggarwal, J., Ryoo, M.: Human activity analysis: A review. *ACM Comput. Surv.* **43**(3), 16:1–16:43 (Apr 2011)
3. Behera, A., Hogg, D.C., Cohn, A.G.: Egocentric activity monitoring and recovery. In: *ACCV* (2012)
4. Blank, M., Gorelick, L., Shechtman, E., Irani, M., Basri, R.: Actions as space-time shapes. In: *ICCV*. pp. 1395–1402 (2005)
5. Cao, Z., Simon, T., Wei, S.E., Sheikh, Y.: Realtime multi-person 2d pose estimation using part affinity fields. In: *IEEE CVPR* (2017)
6. Carsten, O.: From Driver Models to Modelling the Driver: What Do We Really Need to Know About the Driver?, pp. 105–120. Springer London, London (2007)
7. Corporate, S.: State farm distracted driver detection (2016), <https://www.kaggle.com/c/state-farm-distracted-driver-detection>
8. Donahue, J., Hendricks, L.A., Rohrbach, M., Venugopalan, S., Guadarrama, S., Saenko, K., Darrell, T.: Long-term recurrent convolutional networks for visual recognition and description. *IEEE Trans. PAMI* **39**(4), 677–691 (April 2017)
9. Fathi, A., Farhadi, A., Rehg, J.M.: Understanding egocentric activities. In: *ICCV* (2011)
10. Feichtenhofer, C., Pinz, A., Zisserman, A.: Convolutional two-stream network fusion for video action recognition. In: *IEEE CVPR*. pp. 1933–1941 (2016)
11. Girdhar, R., Ramanan, D.: Attentional pooling for action recognition. In: *Advances in NIPS*. pp. 33–44 (2017)
12. Gkioxari, G., Girshick, R., Malik, J.: Contextual action recognition with r*cnn. In: *ICCV*. pp. 1080–1088 (2015)
13. Gupta, A., Davis, L.S.: Objects in action: An approach for combining action understanding and object perception. In: *CVPR* (2007)
14. Heide, A., Henning, K.: The cognitive car: A roadmap for research issues in the automotive sector. *Annual Reviews in Control* **30**(2), 197 – 203 (2006)
15. Herath, S., Harandi, M., Porikli, F.: Going deeper into action recognition: A survey. *Image and Vision Computing* **60**, 4 – 21 (2017)
16. Hssayeni, M., Saxena, S., Ptucha, R., Savakis, A.: Distracted driver detection: Deep learning vs handcrafted features. *Electronic Imaging* (10), 20–26 (2017)
17. Huang, J., Rathod, V., Sun, C., Zhu, M., Korattikara, A., Fathi, A., Fischer, I., Wojna, Z., Song, Y., Guadarrama, S., Murphy, K.: Speed/accuracy trade-offs for modern convolutional object detectors. In: *IEEE CVPR*. pp. 3296–3297 (2017)
18. Jozefowicz, R., Zaremba, W., Sutskever, I.: An empirical exploration of recurrent network architectures. In: *ICML*. pp. 2342–2350 (2015)
19. Kaplan, S., Guvensan, M.A., Yavuz, A.G., Karalurt, Y.: Driver behavior analysis for safe driving: A survey. *IEEE Trans. on Intel. Transp. Syst.* **16**(6), 3017–3032 (Dec 2015). <https://doi.org/10.1109/TITS.2015.2462084>
20. Kim, H.J., Yang, J.H.: Takeover requests in simulated partially autonomous vehicles considering human factors. *IEEE Trans. on Human-Machine Syst.* **47**(5), 735–740 (Oct 2017). <https://doi.org/10.1109/THMS.2017.2674998>
21. Kovashka, A., Grauman, K.: Learning a hierarchy of discriminative space-time neighborhood features for human action recognition. In: *IEEE CVPR* (2010)
22. Laptev, I., Lindeberg, T.: Space-time interest points. In: *ICCV*. pp. 432–439 (2003)

23. Laptev, I., Marszalek, M., Schmid, C., Rozenfeld, B.: Learning realistic human actions from movies. In: CVPR (2008)
24. Liu, J., Luo, J., Shah, M.: Recognizing realistic actions from videos “in the wild”. In: IEEE CVPR. pp. 1996–2003 (2009)
25. Luo, Z., Peng, B., Huang, D.A., Alahi, A., Fei-Fei, L.: Unsupervised learning of long-term motion dynamics for videos. arXiv preprint arXiv:1701.01821 **2** (2017)
26. Mallya, A., Lazebnik, S.: Learning models for actions and person-object interactions with transfer to question answering. In: ECCV. pp. 414–428 (2016)
27. Ng, J.Y.H., Hausknecht, M., Vijayanarasimhan, S., Vinyals, O., Monga, R., Toderici, G.: Beyond short snippets: Deep networks for video classification. In: CVPR (2015)
28. Ranft, B., Stiller, C.: The role of machine vision for intelligent vehicles. IEEE Trans. on Intel. Vehicles **1**(1), 8–19 (2016). <https://doi.org/10.1109/TIV.2016.2551553>
29. Razavian, A.S., Azizpour, H., Sullivan, J., Carlsson, S.: Cnn features off-the-shelf: An astounding baseline for recognition. In: IEEE CVPRW. pp. 512–519 (2014)
30. Rohrbach, M., Amin, S., Andriluka, M., Schiele, B.: A database for fine grained activity detection of cooking activities. In: IEEE CVPR. pp. 1194–1201 (June 2012)
31. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C., Fei-Fei, L.: ImageNet Large Scale Visual Recognition Challenge. IJCV **115**(3), 211–252 (2015)
32. Ryoo, M.S., Aggarwal, J.K.: Spatio-temporal relationship match: Video structure comparison for recognition of complex human activities. In: ICCV (2009)
33. Ryoo, M.S., Rothrock, B., Matthies, L.H.: Pooled motion features for first-person videos. In: IEEE CVPR (2014)
34. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
35. Singh, B., Marks, T.K., Jones, M., Tuzel, O., Shao, M.: A multi-stream bi-directional recurrent neural network for fine-grained action detection. In: IEEE CVPR. pp. 1961–1970 (2016)
36. Singh, D.: Using convolutional neural networks to perform classification on state farm insurance driver images. Tech. rep., Stanford University, Stanford, CA (2016)
37. Tieleman, T., Hinton, G.: Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. COURSERA: Neural networks for machine learning **4**(2), 26–31 (2012)
38. Trivedi, M.M., Gandhi, T., McCall, J.: Looking-in and looking-out of a vehicle: Computer-vision-based enhanced vehicle safety. IEEE Trans. on Intel. Transp. Syst. **8**(1), 108–120 (March 2007). <https://doi.org/10.1109/TITS.2006.889442>
39. Wang, H., Kläser, A., Schmid, C., Liu, C.L.: Dense trajectories and motion boundary descriptors for action recognition. IJCV **103**(1), 60–79 (May 2013)
40. Wang, L., Qiao, Y., Tang, X.: Action recognition with trajectory-pooled deep-convolutional descriptors. In: IEEE CVPR (2015)
41. Wu, Z., Jiang, Y.G., Wang, X., Ye, H., Xue, X., Wang, J.: Fusing multi-stream deep networks for video classification. arXiv preprint arXiv:1509.06086 (2015)
42. Xingjian, S., Chen, Z., Wang, H., Yeung, D.Y., Wong, W.K., Woo, W.c.: Convolutional lstm network: A machine learning approach for precipitation nowcasting. In: Advances in NIPS. pp. 802–810 (2015)
43. Yosinski, J., Clune, J., Bengio, Y., Lipson, H.: How transferable are features in deep neural networks? In: NIPS. pp. 3320–3328 (2014)
44. Yue-Hei Ng, J., Hausknecht, M., Vijayanarasimhan, S., Vinyals, O., Monga, R., Toderici, G.: Beyond short snippets: Deep networks for video classification. In: IEEE CVPR. pp. 4694–4702 (2015)