2018 21st International Conference on Intelligent Transportation Systems (ITSC)
Maui, Hawaii, USA, November 4-7, 2018

# Dilated Convolutional Neural Network for Predicting Driver's Activity

Banafsheh Rekabdar
Department of Computer Science
Southern Illinois University Carbondale
Email: brekabdar@cs.siu.edu

Christos Mousas
Department of Computer Science
Southern Illinois University Carbondale
Email: christos.mousas@siu.edu

***Abstract*—Anticipation and prediction play a pivotal role in human-like perception and memory. Anticipating human activities is an essential capability for any natural and seamless human-robot interaction scenario. To effectively understand and reason about human activities, a smart system needs to have the ability to process sequential/temporal observations that are normally noisy, high dimensional, have long temporal dependencies and have a high frequency (e.g. videos). In this paper, we propose a novel deep learning model architecture to classify driver's actions and activities in real-world scenarios of driving a car in different conditions. Sensory data comes from a variety of sources including a driver facing camera (inside camera), a road facing camera (outside camera), GPS and other car related sensors. The proposed model is flexible and easy to use since it is not relied on external methods to extract key-points from video frames. It uses convolution and max-pooling pairs to understand spatial relationships within video frames and incorporates dilated deep convolutional structures to capture long temporal dependencies, process and predict driver's activities. We show the results to compete with the state of the art in this domain.**

## I. INTRODUCTION

Driving is a necessary element of everyday life. Each year, 1.25 million people are killed on roadways throughout the world. Each day, an estimated 3,400 people are killed globally in road traffic crashes involving cars, buses, motorcycles, bicycles, trucks, or pedestrians [1]. That's why a lot of work has been done to improve driver's assistive technology to generate a more effective warning system to the driver, thus giving driver more time to prevent a critical situation and even prevent accidents from occurring. Recent studies showed that driver's assistive systems which are capable of recognizing intention of the driver are very helpful to provide early warning to avoid/decrease dangerous manoeuvres. Recognizing driver's action and intention is a complex task as it involves multi-dimensional dynamics [8]. Dilated convolutional neural networks (dilated CNN) have recently enjoyed a great success in image segmentation and dense prediction (semantic segmentation), [13], text-to-speech [9], and text classification[10]. In this paper, we address another important application of dilated convolutions – intent recognition and prediction of time-series data.

In this work, we develop a model, based on dilated CNN and convolutional neural networks max-pooling (CNN max-pooling) pairs to predict driver's activities. The key contributions of this work are as follows: 1) We propose a generic architecture for predicting time-series data in robotic application, 2) Our model is able to combine data sources with different characteristics, 3) The model is flexible and easy to use since it is not relied on external methods to extract key-points from video frames, 4) We employed CNN maxpooling to extract high level features automatically, 5) We investigate the use of dilated CNNs for capturing temporal dependencies.

The rest of the paper is organized as follows: in Section II, Section III, Section IV, and Section V we describe the related work, the background overview, the proposed architecture, and the experimental results respectively. We explore future directions and summarize the entire work in Section VI.

## II. PREVIOUS WORK

Activity prediction plays an important role in human-robot interaction/collaboration [11]. Different methods are proposed for activity prediction. Some of them rely on feature matching and Bayesian reasoning [6][12] that normally struggle to learn and correctly model the long temporal aspects of human activities, but other approaches including hidden markov models [4], recurrent neural networks [5], and deep (bidirectional) recurrent neural network [8] are more powerful in modeling temporal relationships. In [2] authors used long-term recurrent convolutional networks for visual recognition and description.

We apply our approach for predicting driver's activities. Several prior work also addressed driver's activities (maneuvers) prediction by using several sensory information like cameras, GPS, and vehicle dynamics [4]. In this paper we used raw inside and outside camera videos, car's speed, and lane information.

The main building block of our work is dilated CNN and CNN max-pooling pairs. The proposed approach is related to the recent work on using RNN-LSTM [5] for maneuver anticipation and using iterated dilated CNN [10] for entity recognition. Our contribution lies in formulating activity prediction in a sensory-fusion deep learning framework using a combination of dilated CNN and CNN max-pooling pairs. In comparison, this work digests the raw data, and it is not dependent on extracting discriminative features as in [4], [5], and [8]. Hence we employ CNN max-pooling pairs to process videos coming from inside (face) and outside (road) cameras. Our method uses dilated CNN to capture the long temporal dependencies unlike other approaches which used

978-1-7281-0323-5/18/$31.00 ©2018 IEEE

LSTM based on RNN and HMM. Dilated CNN supports exponential expansion of the receptive field without loss of resolution or coverage [13] and suffer less from vanishing gradient problem observed in back-propagation through time approaches and they are also easily paralellizable.

## III. Background Overview

The input data for the task of predicting driver's activities, is essentially sequential. This input data usually consists of consecutive video frames captured by inside/outside cameras; GPS information, and car's speed. In this section we reviewed the concepts of convolution operator, recurrent neural network, and dilated convolutions.

### A. Convolution operator

For functions k and f defined on the set $Z$ of integers, the discrete convolution of k and f is given by

$$(k * f)(t) = \sum_{t=-\infty}^{\infty} k(m)g(t-m) \tag{1}$$

The $*$ is referred to the convolution operator. The one-dimensional convolution is an operation between a vector of weights $w \in R^w$ and a vector of inputs viewed as a sequence $seq \in R^{seq}$.

CNNs are widely used in processing sequential data [2]. Three architectural ideas are combined in CNNs to guarantee some order of shift invariance: shared weights, spatial/temporal subsampling, and local receptive fields [7]. Formally, the receptive field of a neuron/node is the area of the original image that can possibly influence the neuron's/node's activation [13]. In CNNs, the size of the receptive field is linearly related to the number of layers and the kernel width. Hence, to cover a longer sequence, a larger receptive field is required. So a larger receptive field requires more layers, and will make training process more difficult. [3]

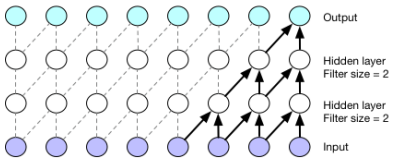


Fig. 1. A stack of convolutional layers.

### B. RNN

Another successful method for working with sequential data is Recurrent Neural Networks (RNNs). Their receptive field could be equal to the entire input. RNNs compute gradients using back-propagation through time. For longer sequences, the gradients should travel long distances [3]. That's why they suffer form the vanishing gradient problem. LSTM is a special kind of RNN which could help in preventing vanishing grandiants by adding a memory to each cell, but they still have trouble learning very long-distance relationships because they need the same number of back-propagation steps as the sequence length [3].

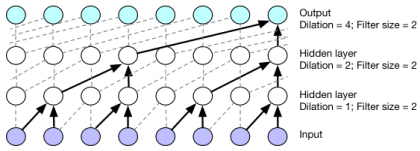


Fig. 2. A stack of dilated convolutional layers for dilations 1, 2, and 4 (Dilated CNN used for inside camera features).

### C. Dilated Convolution operator

For functions k and f defined on the set Z of integers, the l-dilated convolution of k and f is given by

$$(k *_l f)(t) = \sum_{t=-\infty}^{\infty} k(m)g(t-lm) \tag{2}$$

The $*_l$ is referred to l-dilated convolution operator. It is also called a trous, or convolution with holes [9]. The dilated convolution is in the mid point of CNNs and RNNs. This convolution is a way of increasing receptive fields exponentially without loss of resolution or coverage with short-distance gradient propagation [13]. That's why they are very applicable in applications which care more about integrating knowledge of the wider context with less cost. Dilated convolutions also define the spacing between the values in the kernel window (convolution with holes) while in the regular convolutions the kernel window consists of adjacent input.

The kernel window starting at location i of size k with dilation d is defined as follows:

$$[x_i \; x_{i+d} \; x_{i+2d} \;\; x_{i+(k-1)d}] \tag{3}$$

In dilated convolution the receptive field can be expanded exponentially by stacking layers of convolutions with increasingly dilated values, so they have large receptive fields with small number of back-propagarion steps. As a special case, dilated convolution with dilation 1 is equal to the standard convolution. Figure 2 shows dilated convolutions for dilations 1, 2, and 4.

## IV. Proposed Approach

Accurate recognition and precise prediction is a challenging task in robotics. One of the main challenges is that the data comes from multiple sensors, it is normally noisy and high dimensional and it encompasses spatial and temporal relationships. In this paper, we propose a unified framework (see Figure 3) to address the following requirements: I) predicting driver's activity several seconds in advance II) sensory data fusion, III) raw data digestion, IV) capturing the spatial dependencies with CNN max-pooling pairs, and V) understanding temporal relationships in the observations by dilated convolutional component.

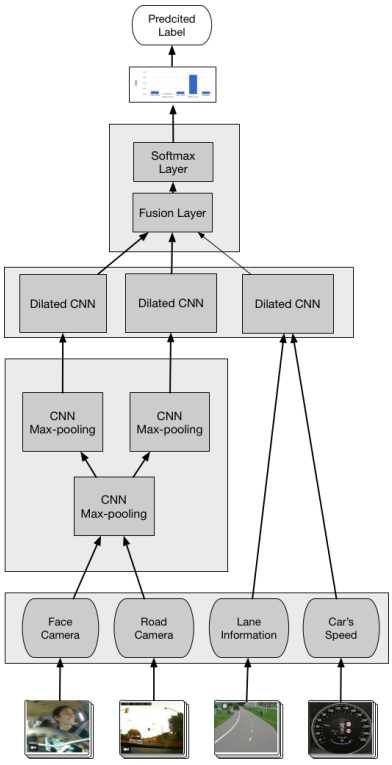


Fig. 3. The proposed model architecture.

### A. Data set

This approach is tested on a driving data set released by [4]. This data set contains recordings from inside and outside of the car. The recordings contain natural driving scenarios with a driver facing camera, a road facing camera, lane information, and the car's speed. The data is collected from 10 drivers in 1180 miles of freeway in the period of 2 month. There are 2 million frames in the videos with various landscape in the complete dataset.

### B. Network architecture

*1) Inputs and Outputs:* The proposed architecture is presented in Figure 3. At each time step, the input is sequences of the input vectors x (face camera), z (road camera), and r (lane information and car's speed) observed so far, and the output is the probability of each class at that time step. The classes for our experiment, consist of 5 maneuvers: right turn, left turn, right lane change, left lane change, and driving straight. For example at time step 1, the input is 3 sequences $(x_1)$, $(z_1)$ and, $(r_1)$, at time step 2, it takes $(x_1, x_2)$, $(z_1, z_2)$, and $(r_1, r_2)$; ...; and, at time step t, the inputs are $(x_1, x_2, ..., x_t)$, $(z_1, z_2, ..., z_t)$, and $(r_1, r_2, ..., r_t)$.

Unlike the other approaches for driver activity prediction [8][4][5], we use raw pixel values as our input features. The CNN max pooling pairs in our architecture learns the visual features from the sensory feeds automatically. Both front and driver facing cameras (inside and outside cameras) captured frames are scaled down to 144 x 96 pixels with three color channels and are sampled at 2 fps frequency.

Car speed readings are available for each camera frame, therefore we sample it alongside the video frames at 2 fps frequency. Road lane information, on the other hand, is provided once for each sample since it wont change within any of the cases covered in our dataset. We normalize all of the input features (pixel values, car speed, lane information) to a 0 to 1 uniform scale in the whole dataset.

*2) Feature Extraction layer:* As extracting high level features from raw sensor outputs (specially cameras) is a tedious and highly specialized task, we use CNN max-pooling pairs to learn these feature representations rather than hand-picking them for the rest of the model. For the first CNN max-pooling component, the weights are shared for both face and road camera feed (inside and outside cameras). Shared weights help the model to learn these low-level visual features from more examples (face and road camera feed together) and these low-level features are normally scene/viewpoint independent. But for learning high-level features, we use separate max-pooling CNNs for face and road cameras since their viewpoints and important key-points are different. This method makes the whole model flexible and easy to use since we do not rely on external methods to extract key-points from video frames.

*3) Dilated CNN layer:* In the next layer, at time step t, the sequence of high level features for face camera $h_1^x, h_2^x, ..., h_t^x$, road camera $h_1^z, h_2^z, ..., h_t^z$ as well as the lane information and car's speed $r_1, r_2, ..., r_t$ are the inputs for 3 dilated convolutional neural networks. In this layer the temporal relationships in the observations are captured.

Dilated CNN computation is shown by the equations 4, 5, and 6. In these equations $F *_{d_i}$ is the $d_i$-dilated convolution in layer $i$ (dilation width is $d_i$). There are $l$ layers of dilated convolutions of exponentially increasing dilation width.

$$c_t^{(1)} = ReLU(F *_{d_1} h_t^x) \tag{4}$$

$$c_t^{(i)} = ReLU(F *_{d_i} c_t^{(i-1)}) \tag{5}$$

$$c_t^{(l)} = ReLU(F *_{d_l} c_t^{(l-1)}) \tag{6}$$

*4) Fusion layer and Softmax layer:* The high level representations from the DCNNs $(c_t^{x,l}; c_t^{z,l}, c_t^{r,l})$ are concatenated and used as input to the fusion layer (equation 7). We chose fusion component to be fully connected because this will let the model to learn inter-sensory data interactions and dependencies, hence the name, fusion layer. The output of the fusion layer $(e_t)$ is passed to the softmax layer for anticipation (equation 8). Softmax is a good choice because the classes are mutually exclusive and the number of choices is not large.

$$e_t = tanh(W_f[c_t^{x,l}, c_t^{z,l}, c_t^{r,l}] + b_f) \tag{7}$$

$$g_t = softmax(W_g e_t + b_g) \tag{8}$$

*5) Loss layer:* We adapt the exponential loss layer proposed by [5] as the base of loss function used in our model. The idea behind the exponential loss is that it will penalize the model for wrong predictions that happens later in time, more than wrong predictions that happen earlier. However, since dilated convolutions needs logarithmic number of back-propagations to reach earlier observations, we propose to use a linear factor in our loss function to achieve the same result. The proposed loss function is presented in the equation 9. N is the number of examples we are calculating the loss for and k is the correct label for the given example. This loss layer guarantees that the model does not over-fit early when not enough data is available.

$$loss = 1/N \sum_{j=1}^{N} \sum_{t=1}^{T} (T-t) \log(g_t^k) \tag{9}$$

### C. Model Training

The networks in our architecture are trained with back propagation using Tensorflow on an Nvidia Geforce GTX 1080 GPU running in Windows box. For training the model, we used adam optimizer. The initial learning rate, beta1, and beta2 are 0.004, 0.0, and 0.999 respectively. Two layer CNN max-pooling pairs used in the shared component; one layer and two layer CNN max-pooling pairs are used in the next component for the inside and outside camera frames respectively. Table I shows the parameters of CNN max-pooling pairs. The overall structure of the CNN max-pooling is shown in the figure 4. Its activation function is ReLU. The fully connected fusion layer uses tanh activation function.

TABLE I
PARAMETERS OF CNN MAX-POOLING; N-IN-CH: NUMBER OF INPUT CHANNELS; N-O-CH: NUMBER OF OUTPUT CHANNELS; MAX-P-S: MAX-POOLING SIZE; FIL-S: FILTER SIZE; M-P: MAX POOLING; IN-CAM: INSIDE CAMERA; OUT-CAM: OUTSIDE CAMERA

| | Fil-s | N-in-ch | N-o-ch | Max-p-s |
|---|---|---|---|---|
| Layer 1-shared CNN-M | 5 x 5 | 3 | 16 | 2 x 2 |
| Layer 2-shared CNN-M | 5 x 5 | 16 | 32 | 2 x 2 |
| Layer 1-CNN- M-p -In-cam | 7 x 7 | 32 | 32 | 2 x 2 |
| Layer 2-CNN- M-p -In-cam | 7 x 7 | 32 | 64 | 2 x 2 |
| Layer 1-CNN- M-p -Out-cam | 5 x 5 | 32 | 32 | 2 x 2 |

A 3 layer dilated CNN with filter size of 2, 2, 2 and dilation width of 1, 2, 4 are used to process the high level road camera (outside camera) features obtained by CNN max-pooling layer. It is shown in Figure **??**. A 3 layer dilated CNN with filter size of 2, 2, 3 and dilation width of 1, 2, 4 are used to process the high level face camera (inside camera) features. This Dilated network is presented in Figure 5. A similar dilated CNN to the one used for face camera is used to process car's speed and lane information. The activation function for the Dilated CNNs are ReLU.

## V. EXPERIMENTS AND RESULTS

We predict maneuvers at every half a second and assign a probability to each of the five maneuvers. The overall driver's activity prediction procedure is shown in Algorithm 1; the threshold value is 0.7.

**Input:** Video frames of inside and outside cameras; car's speed; lane's information; threshold
**Output:** Predicted driver's activity
Initialization;
**while** *t=1 to T* **do**
  Process the input;
  Estimate the probabilities of all the 5 classes ($g_i$);
  v = Maximum ($g_i$)
  **if** *v>Threshold* **then**
    $v_{predicted} = v$
    $t_{early} = T - t$
    break
  **end**
  **return** $v_{predicted}$, $t_{early}$
**end**
**Algorithm 1:** Algorithm of driver's activity prediction

### A. Data and Evaluation setup

The choice of data representation (or features) play a pivotal rule in the performance of machine learning techniques. For the data used in this paper, the sensory input of our system is the sequences derived from the following: (i) outside camera (ii) inside camera (iii) lane information, and (iv) car's speed.

We run our model at every 0.5 seconds in the data feed, where the algorithm processes the recent context and assigns a probability to each of the five maneuvers: left lane change, right lane change, left turn, right turn, driving straight. These five probabilities together form a probability distribution because the output is a softmax.

After anticipation, i.e. when the model computes all five probabilities, the algorithm predicts a maneuver if any of its probabilities is above a certain threshold. If none of the maneuvers probabilities are above this threshold, the algorithm does not make a maneuver prediction. This process repeats every 0.5 seconds and the model makes new predictions at each trial, with the possibility of changing its prediction as more data becomes available over time.

We use the following notations: (i) true predictions (tp): when the predicted maneuver matches the correct maneuver; (ii) false predictions (fp): when the predicted maneuver is wrong; and (iv) missed predictions (mp): when the model does not predict any maneuvers. We evaluate the model using precision and recall metrics:

Precision, is formally defined as the fraction of the predicted maneuvers that are correct and recall is defined as the fraction of the maneuvers that are correctly predicted. For true predictions (tp) we also compute the average time-to-maneuver, where time-to-maneuver is the interval between the start time of the maneuver and the first time model detects it.

$$Pr = tp/(tp + fp + fpp) \tag{10}$$

$$Re = tp/(tp + fp + mp) \tag{11}$$

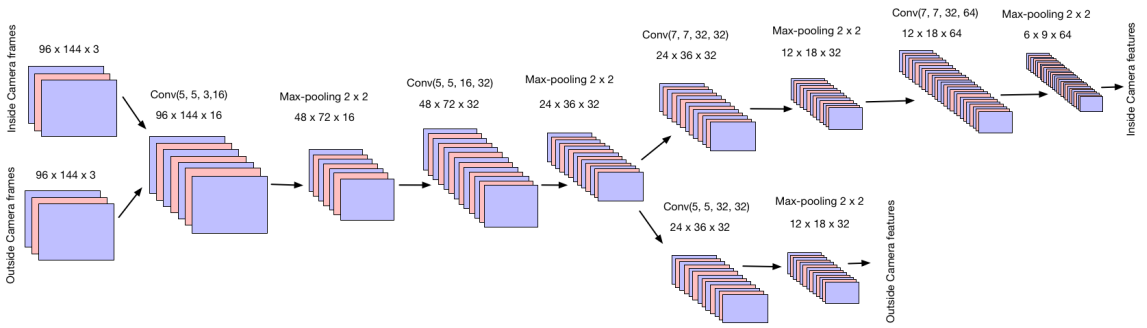


Fig. 4. The structure of CNN max-pooling pairs in the propsoed method

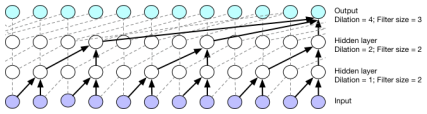


Fig. 5. Dilated CNN used for inside camera features.

TABLE III
PARAMETERS OF FEEDFORWARD NEURAL NETWORK (INITIALIZED BY SDAE WEIGHTS), LR: LEARNING RATE

| LR | Epochs | Batch Size | Hidden Nodes | Output Nodes |
|---|---|---|---|---|
| 0.8 | 100 | 10 | 160 | 5 |

The parameters for all the approaches are chosen as the ones giving the highest F1-score on a validation set. The F1 score is the harmonic average of the precision and recall, where an F1 score reaches its best value at 1 (perfect precision and recall) and worst at 0, defined as follows:

$$F1 = 2/(1/Pr + 1/Re) \tag{12}$$

### B. Comparison with other algorithms

We compare the performance of our proposed method with other deep learning methods like deep CNN, stacked denoising autoencoders (SDAE) and other past work (the results of the following algorithms are obtained from [5] on the features of [4]) and show that our approach, present an advancement over existing models: 1) Chance: Uniformly randomly anticipates a maneuver [5] 2) Support Vector Machine [5], 3) Random-Forest [5] 4) HMM E [5], 5) HMM F [5], 6) HMM E + F [5], 7) IOHMM [4] 8) AIO-HMM [4], 9) S-RNN [5], 10) F-RNN-UL [5], 11) F-RNN-EL [5], 12) F-RNN-EL w/3D head-pose [5].

A three layer stacked denoising autoencoder (SDAE) was created. Each layer is a denoising autoencoder (DAE), having 150 hidden neurons. We trained the layers in a greedy-layer wise fashion. After pre training the SDAE, the upward weights and biases are then used for training a two layer feed forward neural network. Table II and Table III show the SDAE and feed forward neural network parameters.

TABLE II
PARAMETERS OF SDAE

| Learning rate | Epochs | Batch Size |
|---|---|---|
| 0.8 | 100 | 10 |

For the CNN method, all the structure and parameters are similar to the proposed method, the only difference is that instead of three dilated CNN, we used three, five layer CNN. The CNN and SDAE methods are trained by the features proposed by [4]. The details for rest of the approaches are mentioned in [5].

The comparison results of the proposed approach versus other approaches are presented in Table IV. Based on this table our approach performs better than all other methods in-terms of Recall and Precision. All the methods in the Table IV except the proposed one trained with features introduced by [4]. The proposed method is the only approach in Table IV which is flexible and easy to use since it is not relied on external methods to extract key-points from the video frames. This further emphasizes the significance of the proposed approach, which gives very good performance under these situations. Also the time-to-maneuver of the proposed method is 3.75(s) which shows it is the second fastest method to predict the maneuvers.

## VI. CONCLUSION AND FUTURE WORK

We showed the effectiveness of a unified deep learning model in processing streaming data from a wide range of sensors with spatial and temporal dependencies in the context of driver maneuver anticipation. Standard convolution max-pooling pairs were employed to understand raw frames captured from a driver facing and a road facing camera. The kernels in the lower layers of our image understanding component is shared between both cameras, while the top layers have their own trainable kernels for better specialization of face and road features. The next component of our model, uses dilated convolutions to capture the temporal semantics in the sequence of features learned in the lower layers. Finally, we use fully connected layers to fuse representations of different sensors

TABLE IV
COMPARISON OF THE PROPOSED METHOD WITH THE STATE OF THE ART; F-RNN-EL W/3D HP: F-RNN-EL W/3D HEAD-POSE; T: TIME-TO-MANEUVERS

| Methods | Lane change | | | Turns | | | All maneuvers | | |
|---|---|---|---|---|---|---|---|---|---|
| | Pr(%) | Re(%) | T(s) | Pr(%) | Re(%) | T(s) | Pr(%) | Re(%) | T(s) |
| Chance | 33.3 | 33.3 | - | 33.3 | 33.3 | - | 20.0 | 20.0 | - |
| SVM | 73.7 ± 3.4 | 57.8 ± 2.8 | 2.40 | 64.7 ± 6.5 | 47.2 ± 7.6 | 2.40 | 43.7 ± 2.4 | 37.7 ± 1.8 | 1.20 |
| Random-Forest | 71.2 ± 2.4 | 53.4 ± 3.2 | 3.00 | 68.6 ± 3.5 | 44.4 ± 3.5 | 1.20 | 51.9 ± 1.6 | 27.7 ± 1.1 | 1.20 |
| HMM E | 75.0 ± 2.2 | 60.4 ± 5.7 | 3.46 | 74.4 ± 0.5 | 66.6 ± 3.0 | 4.04 | 63.9 ± 2.6 | 60.2 ± 4.2 | 3.26 |
| HMM F | 76.4 ± 1.4 | 75.2 ± 1.6 | 3.62 | 75.6 ± 2.7 | 60.1 ± 1.7 | 3.58 | 64.2 ± 1.5 | 36.8 ± 1.3 | 2.61 |
| HMM E + F | 80.9 ± 0.9 | 79.6 ± 1.3 | 3.61 | 73.5 ± 2.2 | 75.3 ± 3.1 | 4.53 | 67.8 ± 2.0 | 67.7 ± 2.5 | 3.72 |
| IOHMM | 81.6 ± 1.0 | 79.6 ± 1.9 | 3.98 | 77.6 ± 3.3 | 75.9 ± 2.5 | 4.42 | 74.2 ± 1.7 | 71.2 ± 1.6 | 3.83 |
| AIO-HMM | 83.8 ± 1.3 | 79.2 ± 2.9 | 3.80 | 80.8 ± 3.4 | 75.2 ± 2.4 | 4.16 | 77.4 ± 2.3 | 71.2 ± 1.3 | 3.53 |
| S-RNN | 85.4 ± 0.7 | 86.0 ± 1.4 | 3.53 | 75.2 ± 1.4 | 75.3 ± 2.1 | 3.68 | 78.0 ± 1.5 | 71.1 ± 1.0 | 3.15 |
| F-RNN-UL | 92.7 ± 2.1 | 84.4 ± 2.8 | 3.46 | 81.2 ± 3.5 | 78.6 ± 2.8 | 3.94 | 82.2 ± 1.0 | 75.9 ± 1.5 | 3.75 |
| F-RNN-EL | 88.2 ± 1.4 | 86.0 ± 0.7 | 3.42 | 83.8 ± 2.1 | 79.9 ± 3.5 | 3.78 | 84.5 ± 1.0 | 77.1 ± 1.3 | 3.58 |
| SDAE | 70 ± 1.3 | 75.1 ± 2.6 | 2.4 | 76 ± 1.7 | 74.3 ± 1.1 | 2.53 | 71.9 ± 2.1 | 74.8 ± 2.5 | 3.22 |
| Deep CNN | 80 ± 1.9 | 78.1 ± 1.6 | 3.4 | 74 ± 1.2 | 76.3 ± 2.1 | 3.53 | 78 ± 2.0 | 77.5 ± 2.5 | 3.22 |
| F-RNN-EL w/3D hp | - | - | - | - | - | - | 90.5 ± 1.0 | 87.4 ± 0.5 | 3.16 |
| Proposed method | 88.1 ± 0.7 | 88.9 ± 1.9 | 3.61 | 84.1 ± 1.5 | 87.8 ± 3.7 | 3.91 | **91.8** ± 1.0 | **92.5** ± 1.3 | 3.76 |

and learn inter-sensory relationships for the task of driver activity anticipation. We showed that the proposed approach is performing better than the state of the art and requires less hand-picking of features as it works directly with raw sensory data. For future work, we plan on extending this work in a number of directions. First, we plan to implement this architecture on a real-time system that will be used on real-driver's assistive systems. Second, we would like to investigate how the proposed architecture could be applicable in other robotics applications in which prediction is required.

## REFERENCES


[1] Road traffic injuries and deathsa global problem. https://www.cdc.gov/features/globalroadsafety/index.html. Accessed: 2016-11-23.

[2] Jeffrey Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *IEEE conference on computer vision and pattern recognition*, pages 2625–2634, 2015.

[3] Ankit Gupta and Alexander M Rush. Dilated convolutions for modeling long-distance genomic dependencies. *arXiv arXiv:1710.01278*, 2017.

[4] Ashesh Jain, Hema S Koppula, Bharad Raghavan, Shane Soh, and Ashutosh Saxena. Car that knows before you do: Anticipating maneuvers via learning temporal driving models. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3182–3190, 2015.

[5] Ashesh Jain, Avi Singh, Hema S Koppula, Shane Soh, and Ashutosh Saxena. Recurrent neural networks for driver activity anticipation via sensory-fusion architecture. In *Robotics and Automation (ICRA)*, pages 3118–3125. IEEE, 2016.

[6] Kris M Kitani, Yoichi Sato, and Akihiro Sugimoto. Deleted interpolation using a hierarchical bayesian grammar network for recognizing human activity. In *Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, pages 239–246. IEEE, 2005.

[7] Yann LeCun, Yoshua Bengio, et al. Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, 3361(10):1995, 1995.

[8] Oluwatobi Olabiyi, Eric Martinson, Vijay Chintalapudi, and Rui Guo. Driver action prediction using deep (bidirectional) recurrent neural network. *arXiv preprint arXiv:1706.02257*, 2017.

[9] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio. *arXivarXiv:1609.03499*, 2016.

[10] Emma Strubell, Patrick Verga, David Belanger, and Andrew McCallum. Fast and accurate entity recognition with iterated dilated convolutions. In *Conference on Empirical Methods in Natural Language Processing*, pages 2660–2670, 2017.

[11] Zhikun Wang, Katharina Mülling, Marc Peter Deisenroth, Heni Ben Amor, David Vogt, Bernhard Schölkopf, and Jan Peters. Probabilistic movement modeling for intention inference in human–robot interaction. *International Journal of Robotics Research*, 2013.

[12] Zong-Hong Wu, Alan Liu, Pei-Chuan Zhou, and Yen Feng Su. A bayesian network based method for activity prediction in a smart home system. In *Systems, Man, and Cybernetics (SMC)*, pages 001496–001501. IEEE, 2016.

[13] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv:1511.07122*, 2015.