Driver Fatigue Detection using Recurrent Neural Networks

Younes Ed-doughmi
yeddoughmi@gmail.com
Laboratory of Information Processing and Decision Support
BeniMellal, Morrocco

Najlae Idrissi
Laboratory of Information Processing and Decision Support
BeniMellal, Morrocco
n.idrissi@usms.ma

ABSTRACT

Road accidents are becoming a real global scourge because of the high number of victims involved and the severe consequences that affect road users as well as their families.

Despite the awareness campaigns on the vigilance and caution that must be undertaken on the road, deaths caused in road accidents are still increasing and are now considered as a major public health problem, more specifically in Morocco where the roads are among the most deadly.

To address this issue, vehicle manufacturers have made considerable progress in improving the intelligence and capacity of vehicles to perceive and analyze road environments to prevent accidents and secure passengers. However, with all these efforts, accident statistics show that in most cases, accidents are related to the inattention of the drivers and sometimes irresponsible behavior.

Therefore, considerable amount of research has recently been focused on the analysis and study of the general behaviors of drivers on the road, especially somnolence, as it is among the highest risk factors of accidents and is the leading cause of death on roads.

In this paper, we propose a new approach to analyze driver drowsiness by applying a new recurrent neural network architecture to frame sequences of a driver. We used a public data set to train and validate our model and applied a recurrent neural network architecture called "long short-term memory" to detect driver drowsiness.

CCS CONCEPTS

• **Computer systems organization** → **Embedded systems**; *Redundancy*; Robotics; • **Networks** → Network reliability.

KEYWORDS

Driver fatigue detection, drowsiness detection, recurrent neural networks, deep neural networks.

ACM Reference Format:
Younes Ed-doughmi and Najlae Idrissi. 2019. Driver Fatigue Detection using Recurrent Neural Networks. In *Networking, Information Systems Security (NISS19), March 27–29, 2019, Rabat, Morocco.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3320326.3320376

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
NISS19, March 27–29, 2019, Rabat, Morocco
© 2019 Association for Computing Machinery.
ACM ISBN 978-1-4503-6645-8/19/03…$15.00
https://doi.org/10.1145/3320326.3320376

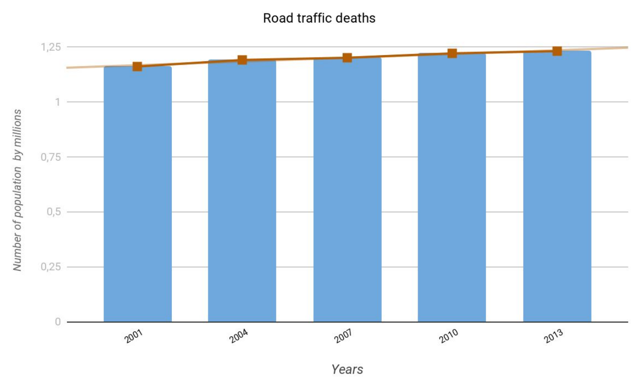

Figure 1: Number of road traffic deaths worldwide – 2015 World Health Organization report [20] –

1 INTRODUCTION

According to the World Health Organization, traffic accidents are a major public health problem worldwide. As stated by the Global status report on road safety 2015 [20], they are one of the leading causes of death and injury.

Each year, nearly 1.25 million people die and millions more are injured or disabled as a result of road accidents. Figure 1 shows the increasing trend in the number of deaths on road from 1.1 to 1.25 million from 2001 to 2013, respectively.

In 2013, a study was conducted in Morocco [8], showing alarming figures of accidents related to drowsiness, which are responsible each year for more than 4000 deaths and estimated material damage worth 14 billion dirhams 2. Drowsiness at the wheel was reported in 36.8% of cases and sleepiness at the wheel in 31.1% of cases, including the quarter during the month preceding the investigation. In addition, 42.4% of the surveyed drivers admitted that they did not respect the recommendations of rest after every 2 h or 500 km of a journey.

Given the seriousness of the problem, which causes social and economic damage in society, the research community and industry believe it is more than necessary to focus on this phenomenon and put in place an effective system to prevent the onset of fatigue and reduce the number of accidents to reduce injuries and death rate in road accidents.

The spectacular advances in the field of deep learning, particularly recurrent neural networks (RNN), have made it possible to provide a solution to this type of problem. In particular, as the data collected via cameras embedded in cars are often heterogeneous and voluminous, it corresponds well to the performance of these

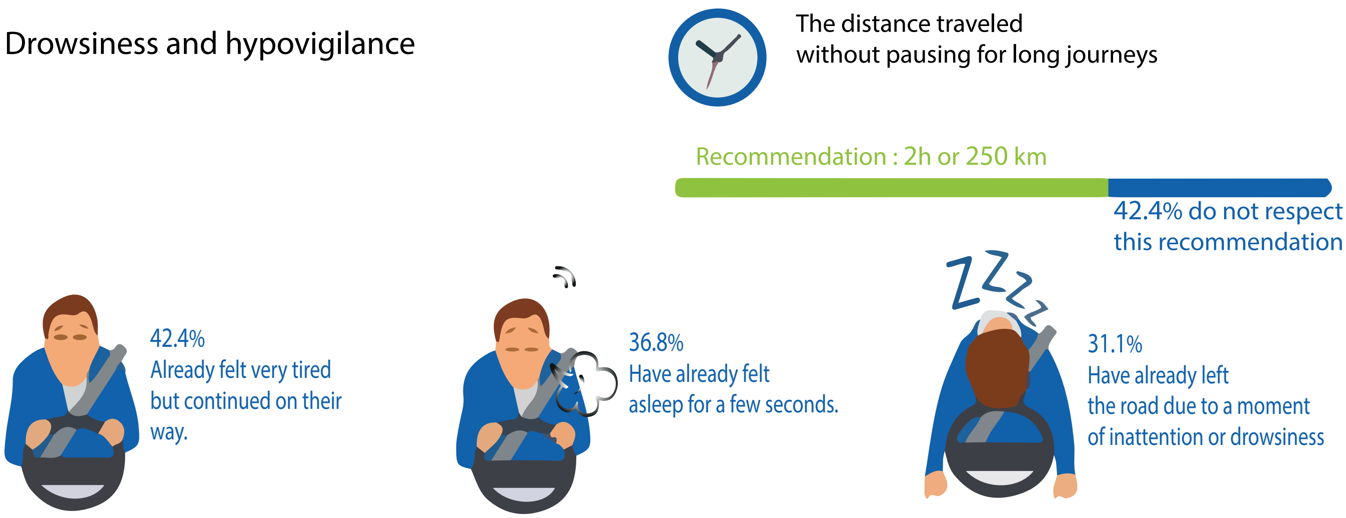

Figure 2: Study of the prevalence and risk factors for drowsy driving in a Moroccan population

calculation models, which are powerful and have proved promising in many arduous tasks.

The idea of RNN is to use sequential information and execute the same task for each element of a sequence from where the name has recurred. The RNN network training techniques are the same as those for classical networks (Backpropagation [12]), except that RNN use memory to capture the state of events calculated in the past. Thus, the RNN allow us to solve the problem of driver drowsiness, and calculate the sequences of the onset of drowsiness, thus allowing the system to alert the driver a few seconds before the loss of alertness in a definitive manner.

For this, we used the dataset of [6] to implement the method used in [27]. This method aims to improve the accuracy of drowsiness detection through binary classification, as the dataset contains several classes related to drowsiness, with a single gray layer. This allows us to perform supervised automatic learning calculations on the drowsiness part to achieve very optimal learning with a higher success rate. The splitting of the video scenes of the dataset to short durations of 7 s at the maximum is necessary to choose the scenes that reflect the action of drowsiness.

After the preparation of the dataset, the learning was initiated using the algorithm detailed in the following sections. Next, some other calculations were performed using additional algorithms to realize a statistical table to present a global idea on the results of the different algorithms that can be used in this type of problem.

In summary, the objective of this work is to improve road safety conditions to anticipate this type of fatal accident. Therefore, we propose RNN algorithms that have already proved their worth in other fields similar to our problem to detect an action based on a sequence of images. This could enable the automotive industries to use this type of study to start feasible and realizable projects.

2 RELATED WORK

In this section, we present an overview of the previous studies conducted on the detection of drowsiness. Various methods have been proposed to alert or detect certain signs of drowsiness. These methods are based on the driver's behavior, appearance, or both, producing a precise result at time t for most of these methods. However, we do not know what this behavior is related to in the case of drowsiness at t_{-1} compared with a case of normal driver behavior. To solve this problem, we used the RNN calculation models, which integrate time parameter (t) in their calculation complexity.

2.1 Detection of driver fatigue according to the state of the eye

Studies conducted on this type of problem [17] [2] [19] in general present a system based on video surveillance via a camera positioned in front of the driver to detect and calculate the frequency of eye blinks and monitor the level of fatigue. These studies used a cascade of classifiers based on Adaboost for rapid detection of the eye area.

In summary, this technique uses a Haar-like descriptor [18] and an AdaBoost classification algorithm [10] for facial and eye detection by using Percent Eye closure (PERCLOS) to assess driver fatigue. PERCLOS assesses the proportion of the total time that the drivers' eyelids are ≥80% closed and reflects slow eyelid closure rather than blinking.

2.2 Pose estimation of the driver's head

The separate head-pose estimation system is a technique that induces the orientation of the head relative to a camera view [3]. Therefore, the lens is detected if the driver tilts his/her head forward, indicating loss of consciousness.

For this, a new descriptor was used in this study that was obtained by merging four descriptors of the most relevant head orientation: the steerable filters, histogram of oriented gradients, Haar features, and an adapted version of the Speeded-Up Robust-Features descriptor. Then, these characteristics were used in the support vector machine classification algorithm [4].

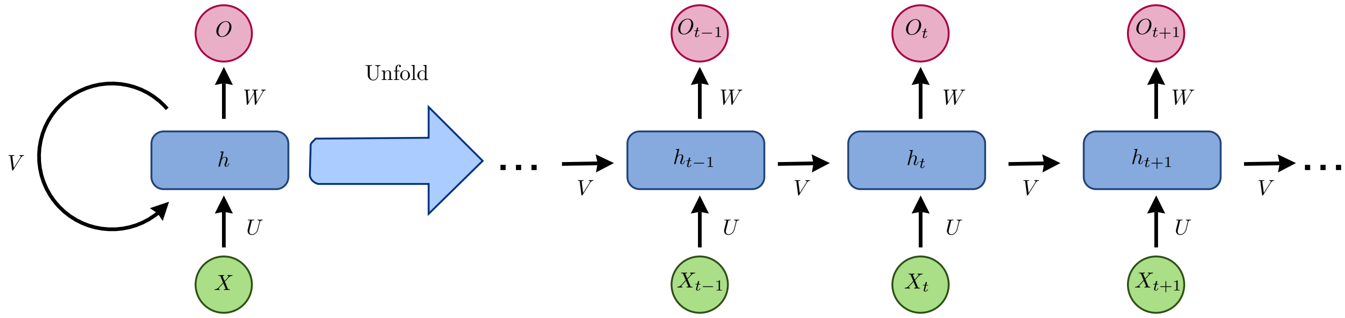

Figure 3: Recurrent neural network and the unfolding in the computation time involved in its forward computation.

2.3 Driver behavior detection by a system based on KINECT

The proposed solution is a system that combines fatigue detection and undesirable driver-behavior detection while driving [24]. Next, the system activates a signal by using the Kinect developer toolbox for Windows [28].

Recent advances in 3D depth cameras, such as Microsoft Kinect sensors, have created many opportunities for a professional enthusiast or developer to detect simple postures, track gestures, calculate relative positions and angles between a person's joints, and convert them into a format that developers can exploit to create new experiences.

2.4 Deep learning neural networks

Deep learning is a growing trend in general data analysis and has been termed one of the 10 breakthrough technologies of 2018 [21].

Deep learning is an improvement of artificial neural networks, and consists of more layers that permit higher levels of abstraction and improved predictions from data [22] [11]. Deep learning is still emerging as the leading machine-learning tool in the general imaging and computer vision domains.

3 PROPOSED APPROACH

This section presents the proposed network architecture. The basic architecture is described first, and then a detailed description of the layers chosen for the two solutions is provided.

3.1 RECURRENT NEURAL NETWORKS

Recent advances in deep learning have made it possible to extract high-level features from raw sensory data, leading to breakthroughs in computer vision [23].

RNNs are inherently deep in time as their hidden state is a function of all previous hidden states Figure 3.

The simplest recursive network comprises a repeating unit connecting an input vector $(x_1, x_2..., x_T)$, a hidden layer vector of vectors $(h_1, h_2, ..., h_T)$, and an output vector $(y_1, y_2, ..., y_T)$. The representations are generated by the nonlinear transformation of the input sequence from $t = 1$ to T.

Some simple recurrent networks (SRNs) include the Elman network [9] and Jordan network [13], which are described in the following equations:

- Elman network

$$h_{\mathrm{x}} = \sigma_{\mathrm{h}}(Wx_{\mathrm{t}} + Uh_{\mathrm{t-1}} + b_{\mathrm{h}}) \tag{1}$$
$$y_{\mathrm{t}} = \sigma_{\mathrm{o}}(Wh_{\mathrm{t}} + b_{\mathrm{o}}) \tag{2}$$

- Jordan network

$$h_{\mathrm{x}} = \sigma_{\mathrm{h}}(Wx_{\mathrm{t}} + Uy_{\mathrm{t-1}} + b_{\mathrm{h}}) \tag{3}$$
$$y_{\mathrm{t}} = \sigma_{\mathrm{o}}(Wh_{\mathrm{t}} + b_{\mathrm{o}}) \tag{4}$$

where W and U are the parameter matrices, b is the vector, and σ_{h} and σ_{o} are the activation functions.

In this section, we discuss the two most used RNN architectures, which can be an optimal solution to the problem discussed in this work.

3.2 Long Short-term Memory Cell

The long short-term memory (LSTM) block or network is a simple RNN, which can be used as a building component or block (of hidden layers) for an eventually bigger RNN. The LSTM block is itself a recurrent network because it contains recurrent connections similar to connections in a conventional RNN [26].

LSTM consists of three gates, namely, input gate i, output gate o, and forget gate f, as well as memory cell c. At each time step t, LSTM first computes gate activations i_t, f_t(6)(5) and updates the memory cell from c_{t-1} to c_t (8). It then computes the output gate activation o_t (7), and finally outputs hidden representation h_t (9). The inputs into the LSTM are observations x_t and the hidden representation from the previous time step h_{t-1}. LSTM applies the following set of update operations:

$$f_t = \sigma_g(W_f * x_t + U_f * h_{t-1} + V_f \circ c_{t-1} + b_f) \tag{5}$$
$$i_t = \sigma_g(W_i * x_t + U_i * h_{t-1} + V_i \circ c_{t-1} + b_i) \tag{6}$$
$$o_t = \sigma_g(W_o * x_t + U_o * h_{t-1} + V_o \circ c_{t-1} + b_o) \tag{7}$$
$$c_t = f_t \circ c_{t-1} + i_t \circ \sigma_c(W_c * x_t + U_c * h_{t-1} + b_c) \tag{8}$$
$$h_t = o_t \circ \sigma_h(c_t) \tag{9}$$

3.3 Architectures

The overall architecture of the proposed drowsiness detection model is based on the LSTM modele. We have tried to establish the best profile from this calculation model, as it is recognized as one of

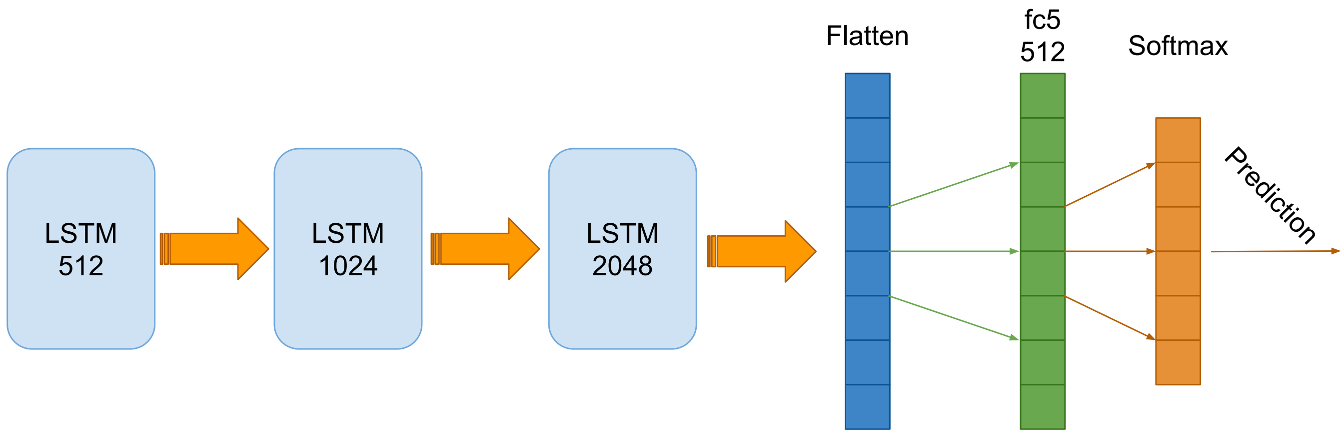

Figure 4: LSTM composite model.

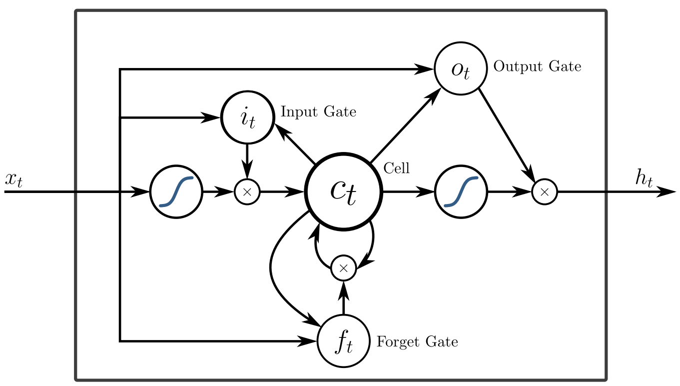

Figure 5: Long Short-term Memory Cell

the most accurate models for similar problems. The following paragraphs describe the architecture and layers used in deep learning.

The LSTM composite model [25] is less deep and reflects the speed of training; however, it utilizes a considerable amount of time to converge on to a good result from several epochs. Therefore, for our proposed architecture, we used three LSTM layers with an increase in neurons from one layer to another, after a flattened layer is applied before the fully connected layer, which is followed by a softmax output layer that gives for the prediction of the traning (Figure 4).

The Softmax function, which is the most well-known and widely-accepted version of the softmax function is represented in the following studies: [11].

4 EXPERIMENTS

4.1 DATASET

For learning driver drowsiness, we used a drowsiness-detection dataset. This dataset containing 18 subjects with 5 scenarios (BareFace, Glasses, Night_BareFace, Night_Glasses, Sunglasses) The sequences corresponding to two most important scenarios, that is, the combination of drowsiness-related symptoms (yawning, nodding, slow blink rate) and non-drowsiness-related actions (talking, laughing, looking at both sides), were recorded for 1.5 min. (see figure 6), which is a different scenario. The figure allows us to study all possible cases for drowsiness of a driver.

In this way, we cut all the videos according to the desired behavior to obtain the sequences of the sleepiness class only. We extracted two clips from each video, and each of these clips lasts at most 7 s. Next, we prepared another subdataset that corresponds to the normal behavior of the driver. At the end of our cutting, we obtained 849 clips.

This cutting of videos from the dataset was preferred to better extract sequences that reflect drowsiness and avoid sequences that can distort the learning results.

4.2 Training

The training of the dataset begins by cutting videos of 36 subjects, as described in the previous subsection, to create two learning classes such that the first class corresponds to drivers' normal behavior and the second class corresponds to the drivers' sleepiness behavior. All videos were divided into three categories: 60% for training, 30% for validation, and 10% for testing.

Then, depending on the chosen architecture, the image sizes of each clip were resized to optimize the memory size of the machine.

To realize this training, we used a PC with Alienware R17, Ubuntu 16.04 LTC, 8G GPU, and 16G RAM. The software level is based on the Keras [5] framework with the python development script, and is a model-level library, providing high-level building blocks for developing deep learning models by using TensorFlow [1] backend.

At the end of the learning, a test phase was realized on the clip sets at the beginning of the cutting to validate the efficacity of the generated model. Table 1 shows the results of the learnings realized on several calculation models compatible with the learning of the video sequences.

5 DISCUSS RESULTS

In this section, we present the training results of each architecture in the graph form, with the acceptance rate converging toward 92%

Figure 6: Different class sequences to use in order as we perform the detection:
- Driver with glasses.
- Driver without glasses in the night.
- Driver with glasses in the night.
- Driver without glasses in the day.
- Driver with sunglasses.

Table 1: Experimental results on the somnolence dataset.

Model	Accuracy	Validate	Test
LSTMs	**92.71 %**	**91.41 %**	73.43 %
LSTM-CNN [7, 15]	92.04 %	85.86%	73.18 %
MLP [16]	71.18 %	72.22 %	60.90 %

on 100 epochs, which uses several layers in depth compared to the LSTM architecture.

Table 1 presents the result of our work on different calculation models with the accuracy rates and validation, as well as the test rate performed on clips not used in the learning step.

Figure 7 reflects the learning evolution realized by the LSTM architecture.

6 PROPOSITION SYSTEME

In this section, we propose a software architecture for a mobile application illustrated in figure x, in which the mobile application will use the phone's camera to capture a sequence of frames at a frequency of 5 frames a second, and through the drowsiness prediction model, computations will be performed in real-time, if the model predicts drowsiness, a visual and audio message will be run on the user's phone.

In a second step, and with the agreement the user, for each detected prediction, a feedback message will be requested to the user to label the sequences of images used. Then, a transfer is executed to a web server, which retrieves the labelled images, these images will be saved and analyzed for the prediction module to be trained. The objective of this step is to ensure that the system's maintenance is scalable and to improve the accuracy of the model's prediction.

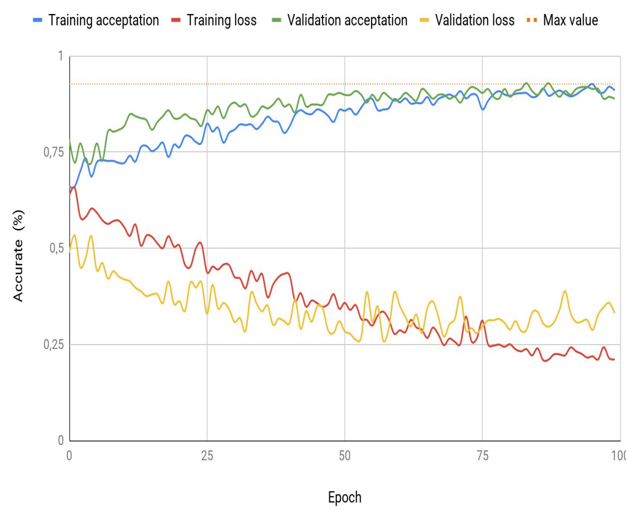

Figure 7: Result training for LSTM.

The technical stack proposed to implement our software architecture is described below:

- Mobile application: Kivy and tensorflow (Python3)
- Web server: Flask and Tensorflow (python3)
- Database: Postgresql (sql)
- Data storage: File system or cloud.

7 CONCLUSIONS

In this study, we attempted to show the possibility of using deep learning architectures on RNNs by using LSTM trained on large-scale video datasets. This architecture respond well to the sequential

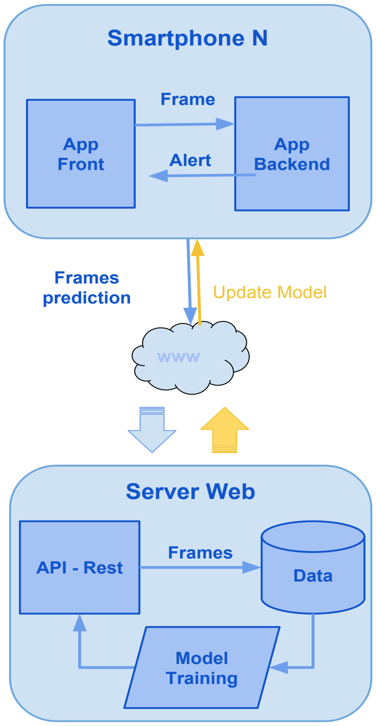

Figure 8: Software architecture for a mobile application and web server

problem. In addition, this method have well shown their effectiveness on subjects close to our problem.

Moreover, our study is based on an action that lasts over time and requires a change in the driver's posture. This is considered as a limit for cases in which the driver is extremely drowsy but does not change posture. This type of sequence cannot be detected using the proposed method; therefore, we can strengthen this study by associating it with studies conducted on the physiology of the driver [14].

REFERENCES

[1] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. 2016. TensorFlow: A System for Large-Scale Machine Learning.. In *OSDI*, Vol. 16. 265–283.

[2] AMER Al-Rahayfeh and Miad Faezipour. 2013. Eye tracking and head movement detection: A state-of-art survey. *IEEE journal of translational engineering in health and medicine* 1 (2013), 2100212–2100212.

[3] Nawal Alioua, Aouatif Amine, Alexandrina Rogozan, Abdelaziz Bensrhair, and Mohammed Rziza. 2016. Driver head pose estimation using efficient descriptor fusion. *EURASIP Journal on Image and Video Processing* 2016, 1 (2016), 2.

[4] Christopher JC Burges. 1998. A tutorial on support vector machines for pattern recognition. *Data mining and knowledge discovery* 2, 2 (1998), 121–167.

[5] François Chollet et al. 2015. Keras.

[6] National Tsuing Hua University Computer Vision Lab. 2016. Driver Drowsiness Detection Dataset. http://cv.cs.nthu.edu.tw/php/callforpaper/datasets/DDD/.

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

[8] M El Ftouh, A Derradji, A Jniene, and M-T El Fassy Fihry. 2013. Étude de la prévalence et les facteurs de risque de la somnolence au volant dans une population marocaine. *Médecine du Sommeil* 10, 4 (2013), 141–145.

[9] Jeffrey L Elman. 1990. Finding structure in time. *Cognitive science* 14, 2 (1990), 179–211.

[10] Yoav Freund, Robert E Schapire, et al. 1996. Experiments with a new boosting algorithm. In *icml*, Vol. 96. 148–156.

[11] Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. 2016. *Deep learning.* Vol. 1. MIT press Cambridge.

[12] Robert Hecht-Nielsen et al. 1988. Theory of the backpropagation neural network. *Neural Networks* 1, Supplement-1 (1988), 445–448.

[13] Michael I Jordan. 1997. Serial order: A parallel distributed processing approach. In *Advances in psychology.* Vol. 121. Elsevier, 471–495.

[14] Victor Javier Kartsch, Simone Benatti, Pasquale Davide Schiavone, Davide Rossi, and Luca Benini. 2018. A sensor fusion approach for drowsiness detection in wearable ultra-low-power systems. *Information Fusion* 43 (2018), 66–76.

[15] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems.* 1097–1105.

[16] PS Jagadeesh Kumar. 2018. Multilayer Perceptron Neural Network Based Immersive VR System for Cognitive Computer Gaming. In *Progress in Advanced Computing and Intelligent Engineering.* Springer, 91–102.

[17] Yuriy Kurylyak, Francesco Lamonaca, and Giovanni Mirabelli. 2012. Detection of the eye blinks for human's fatigue monitoring. In *Medical Measurements and Applications Proceedings (MeMeA), 2012 IEEE International Symposium on.* IEEE, 1–4.

[18] Rainer Lienhart and Jochen Maydt. 2002. An extended set of haar-like features for rapid object detection. In *Image Processing. 2002. Proceedings. 2002 International Conference on*, Vol. 1. IEEE, I–I.

[19] Bappaditya Mandal, Liyuan Li, Gang Sam Wang, and Jie Lin. 2017. Towards detection of bus driver fatigue based on robust visual analysis of eye state. *IEEE Transactions on Intelligent Transportation Systems* 18, 3 (2017), 545–557.

[20] World Health Organization. 2015. *Global status report on road safety 2015.* World Health Organization.

[21] MIT Technol. Rev. 2018. Dueling Neural Networks. https://www.technologyreview.com/lists/technologies/2018/. [Online; accessed 03-Mars-2018].

[22] Jürgen Schmidhuber. 2015. Deep learning in neural networks: An overview. *Neural networks* 61 (2015), 85–117.

[23] Pierre Sermanet, Koray Kavukcuoglu, Soumith Chintala, and Yann LeCun. 2013. Pedestrian detection with unsupervised multi-stage feature learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.* 3626–3633.

[24] Cleshain Solomon and Zenghui Wang. 2015. Driver Attention and Behavior Detection with Kinect. *J. Image Graph* 3, 2 (2015), 84–89.

[25] Nitish Srivastava, Elman Mansimov, and Ruslan Salakhudinov. 2015. Unsupervised learning of video representations using lstms. In *International Conference on Machine Learning.* 843–852.

[26] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems.* 3104–3112.

[27] Ching-Hua Weng, Ying-Hsiu Lai, and Shang-Hong Lai. 2016. Driver Drowsiness Detection via a Hierarchical Temporal Deep Belief Network. In *Asian Conference on Computer Vision.* Springer, 117–133.

[28] Zhengyou Zhang. 2012. Microsoft kinect sensor and its effect. *IEEE multimedia* 19, 2 (2012), 4–10.