



HAL
open science

Driver Drowsiness Detection Model Using Convolutional Neural Networks Techniques for Android Application

Rateb Jabbar, Mohammed Shinoy, Mohamed Kharbeche, Khalifa Al-Khalifa,
Moez Krichen, Kamel Barkaoui

► To cite this version:

Rateb Jabbar, Mohammed Shinoy, Mohamed Kharbeche, Khalifa Al-Khalifa, Moez Krichen, et al.. Driver Drowsiness Detection Model Using Convolutional Neural Networks Techniques for Android Application. ICIoT 2020, 2020, Doha, Qatar. pp.237-242, 10.1109/ICIoT48696.2020.9089484 . hal-02479367

HAL Id: hal-02479367

<https://hal.science/hal-02479367v1>

Submitted on 31 Oct 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Driver Drowsiness Detection Model Using Convolutional Neural Networks Techniques for Android Application

Rateb Jabbar^{*†}, Mohammed Shinoy^{*}, Mohamed Kharbeche^{*},
Khalifa Al-Khalifa[§], Moez Krichen[‡], Kamel Barkaoui[†]

^{*}*Qatar Transportation and Traffic Safety Center, Qatar University, Qatar*
{rateb.jabbar,m.shinoy,mkharbec}@qu.edu.qa

[§]*College of the North Atlantic-Qatar, Doha, Qatar*
alkhalifa@cna-qatar.edu.qa

[†]*Cedric Lab, Computer Science Department, Conservatoire National des Arts et Meteirs, France*
{jabbar.rateb.auditeur,kamel.barkaoui}@cnam.fr

[‡]*ReDCAD Laboratory, National School of Engineers of Sfax, University of Sfax, Tunisia*
moez.krichen@redcad.org

Abstract—A sleepy driver is arguably much more dangerous on the road than the one who is speeding as he is a victim of microsleeps. Automotive researchers and manufacturers are trying to curb this problem with several technological solutions that will avert such a crisis. This article focuses on the detection of such micro sleep and drowsiness using neural network-based methodologies. Our previous work in this field involved using machine learning with multi-layer perceptron to detect the same. In this paper, accuracy was increased by utilizing facial landmarks which are detected by the camera and that is passed to a Convolutional Neural Network (CNN) to classify drowsiness. The achievement with this work is the capability to provide a lightweight alternative to heavier classification models with more than 88% for the category without glasses, more than 85% for the category night without glasses. On average, more than 83% of accuracy was achieved in all categories. Moreover, as for model size, complexity and storage, there is a marked reduction in the new proposed model in comparison to the benchmark model where the maximum size is 75 KB. The proposed CNN based model can be used to build a real-time driver drowsiness detection system for embedded systems and Android devices with high accuracy and ease of use.

Index Terms—Driver Behaviour Monitoring System, Drowsiness Detection, Real-Time Deep Learning, Convolutional Neural Networks, Facial Landmarks, Android.

I. INTRODUCTION

The National Highway Traffic Safety Administration (NHTSA) statistics show that about 2.5% of all fatalities during a crash is attributed to drowsy driving [1]. In 2015 alone, the total number of crashes related to drowsy driving was over 72,000. Remarkably, vehicle crashes caused by impaired driving are much more frequent than those caused by intoxicated drivers [2]. People who drive under the influence are just slower in their reaction time when compared to normal active drivers. Drowsy driving, on the other hand, makes the drivers a victim of microsleeps.

As nearly 1.4 million people die on the road every year thereby making it the 7th global cause of death in 2016 [3], it is no wonder that the automotive industries, research institutions, and other governmental agencies are creating technologies that prevent such scenarios. Among the many crash aversion techniques and other safety features that are developed in newer cars, there is a lot of interest on detecting fatigued and drowsy drivers i.e by using cameras, sensors and other instruments to alert and thereby avert fatal crashes.

Automotive companies such as Mercedes-Benz [4], Tesla [4], and others have their forms of driver assistance systems. Such as variable cruise control, automatic braking systems, lane departure warnings and assisted steering. These technologies have helped drivers to stop crashes from happening. Recently Samsung has partnered with Eyesight [5] to track a drivers attention by reading facial features and patterns. Such a tracking system can warn drivers how attentive they are while driving. Although these advancements are there most of these systems are proprietary and limited to high end vehicles.

There has been a significant rise in the number of vehicles that are equipped with Android Auto or Apple Car nowadays. Most of the cars that are being introduced now have these components built-in. Such features are now readily available in lower-end cars too. Due to this, drowsiness detection systems can be easily developed around such built-in Android and iOS platforms. Embedded devices or mobile phones that can easily pair with these car dashboards can be used to enhance driver behavior detection using simple camera setup and state of the art computer vision systems powered by Deep Learning.

Microsleeps are of short duration where the driver has his eyes closed and is not perceiving any visual information and thereby cannot react if the car departs from the lane or if the car comes to halt due to harsh braking. With the advent

of new sensors and radars in high-end cars, cars can notify or even take action to avoid such crashes. Although this is a great advancement, it would also be beneficial if the car knows that the driver is sleepy and requests him to take rest. Also, another common issue with the current machine vision models is the fact that most of these algorithms are bulky (large in size) and require dedicated hardware to run the models that have been developed. They do not run efficiently on devices with low computational power. This paper details such a Deep Learning-based drowsy detection algorithm that can potentially enable such an intervention. Moreover, it is simple and easy to configure on any mobile or embedded device.

The rest of the paper is divided as follows. In the next section II, the literature review, a few of the state-of-the-art systems for driver drowsiness detection systems are presented. In section Solution and Methodology III, the proposed algorithm and the methodology based on CNN and Facial Landmark Detection (D2CNN-FLD) are described. A discussion regarding the implementation of an Android system with this algorithm is also explained. Furthermore, the accuracy, efficiency, and comparisons of the given model are detailed in Experimental Results in IV. Finally, conclusions V and possible avenues to be investigated in the future are presented.

II. LITERATURE REVIEW

As sleep-induced crashes represent a significant portion of vehicle crashes in the world, researchers and automotive companies have created different solutions ranging from finding patterns in driving habits to analyzing brain waves and vitals of the driver while driving. Most of these solutions are backed by some predictive algorithms powered by statistics and machine learning. The most common ones can be broadly divided into three categories as described in the following paragraphs.

One approach is to find changes in vehicle behavior, such as the one proposed by McDonald et al. [6]. He created a contextual and temporal algorithm that utilizes the steering angle, vehicle speeds, and accelerator pedal positions. These values are passed into a Bayesian Network which determines if a driver shows characteristics of drowsy behavior. The algorithm was found to have lower false-positive rates than PERCLOS [7] methods, which predicts drowsiness based on eyelid movements and patterns. The takeaway from this study was that to predict correctly, the context of the situation is crucial. The data that it captures over a previous 10-second period is vital in understanding whether the person is at risk of drowsiness related lane departures.

A second approach is based on studies focused on using the drivers' vitals, brain waves, and readings from Electro-Encephalo Grams (EEG's) to make predictions. Wei et al. [8] made comparisons between non-hair bearing EEG Brain-Computer Interfaces that are easy to wear and less intrusive than the lab-based whole scalp EEG's which are less comfortable. The study showed that non-hair bearing devices had no significant reduction in performance when compared to whole-scalp EEG. Thereby with this finding one may develop less intrusive and comfortable headbands. EEG alone is unable to detect all stages of drowsiness, so Kartsch et al. [9] used EEG with Inertial Measurements Units (IMU) sensors to detect

5 levels of drowsiness with about 95% accuracy. The team fused behavioral information from IMU and EEG information to detect drowsiness. Another drawback of the EEG system was the power requirements of these devices. Their technology has also facilitated the implementation of a parallel ultra-low power (PULP) platform on a microcontroller which extended the battery life to almost 46 hours, thereby creating devices that are always wearable and require low maintenance. Tateno et al. [10] developed a system that just uses heart rate monitoring to detect the respiration of a person and thereby detect drowsiness. The methodology was found to be an effective predictor for respiration and thereby drowsiness.

Yet another technique is to utilize the power of computer vision. The recent breakthroughs in Deep Learning have provided new tools to computer vision for detection and classification. Computer vision-related applications utilize these methods in object detection, health and wellness, and even agricultural applications too [11]. A big impact in this space has been for imaging data. Since drivers facial features change significantly once he gets tired, computer vision scientists have attempted to capitalize on this and use it to provide solutions for drowsiness detection. Tayab Khan et al. [12] proposed a solution to measure the angle of eyelid curvature and thereby identify if the eyes are closed or not. They achieved 95% accuracy with this method, but the limitation is that there needs to be enough light for the method to work as it functions poorly at night. Shakeel et al. [13] used MobileNet-SSD architecture to train a custom dataset of 350 images. The model was capable of achieving a Mean Average Precision of 0.84. The system was cost-effective and efficient as the algorithm could be deployed in an Android device and the camera stream could be classified in real-time. Celona et al. [14] proposed a vision-based Multi-Task Driver Monitoring Framework that analyzes eyes, mouth, and the pose of the head simultaneously to predict the level of the drowsiness. This study was conducted on the NTHU [15] dataset. Another study conducted by Xie et al. [16] used transfer learning and sequential learning from yawning video clips to detect yawning on the YawDD and NTHU-DDD database. This system was able to have higher precision and was robust to changes in the position and angle of the face to the camera. Mehta et al. [17] developed an Android application that is capable of detecting facial landmarks and then computing the Eye Aspect Ratio (EAR) and Eye Closure Ratio (ECR) to predict drivers drowsiness based on machine learning models with an accuracy of 84%.

A method that many companies are trying to follow is to combine the three different techniques explained above and utilize multiple inputs to come to a decision. A start-up Ellcie-Healthy [18] has developed a smart glass that embeds a drowsiness detection application by including blink detection, eye tracking, and vitals monitoring. The smart glass monitors these inputs and provides drowsiness intervention by beeping and thereby asking the driver to take a rest. The combination techniques require multiple sensors such as infrared, cameras and heart rate monitors on a single platform to provide great results. However, these instruments are quite expensive and require proprietary solutions to be set up.

All these innovations in deep learning space come at a



Fig. 1. NTHU Dataset Sample Images

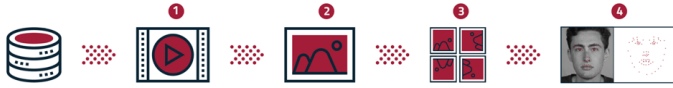


Fig. 2. Video preprocessing outline

cost, a bigger model with higher computational requirements. This is the bane of deep learning. In our previous work [19] [20] a machine learning model that classified drowsy images using a multi-layer perceptron-based model [D2MLP-FLD] was proposed. The result was a respectable accuracy of 81% while creating a model that was 100KB in storage size. The idea that is proposed in this paper is to improve upon the algorithm and create a lightweight application in terms of performance as well as storage space.

III. SOLUTION AND METHODOLOGY

A. Dataset and Preprocessing

The Deep Learning model developed here is trained on videos obtained from the National Tsing Hua University (NTHU) Driver Drowsiness Detection Dataset [15]. These videos are preprocessed to create images for this study.

1) *NTHU Dataset*: This dataset contains 22 subjects from different ethnicities who have been recorded under day and night conditions. The driving situations and behavior were simulated, which included conversations, yawning, slow rate blinking, sleepy head movements, and drowsy eyes as shown in 1. The recording was done using infrared cameras to obtain night-time videos too. The resulting content is 9.5 hours of videos that have a resolution of 640*480 at 30 frames per second containing different subjects enacting regular and drowsy driving behaviors.

2) *Preprocessing the dataset*: Video preprocessing happens in 4 stages as shown in the Figure. 2:

- Step 1 Extracting Images from Video Frames:
As we cannot train the neural network on video data, the images need to be extracted and then fed into the model. The video specifications are at 30 frames per second. These frames are converted to images and thereby correspond to a total of around 600,000 images.



Fig. 3. Data Augmentation

- Step 2 Data Augmentation:
In the Deep Learning training phase, it is important to have more data so that the model can learn all the nuances and variations in the images. A common method to increase the number of training points is to use data augmentation. Codebox [21] was used to generate new images by performing a set of augmentation operations on the images extracted from video frames, as illustrated in Figure 3.
- Step 3 Extracting landmark coordinates from images:
In order to locate and represent salient parts of the face such as eyes, mouth, nose and the jawline, facial landmarks are used. These landmarks are essential for head pose estimation, blink detection, yawning detection, etc. An open-source C++ library, Dlib [22] has pre-written functions that are used to obtain facial landmarks. This library is programmed to find the x, y coordinates of 68 facial landmarks to map the facial structure as shown in (Figure 4).
Dlib Library utilizes the power of OpenCV's Haar Cascades to detect facial landmarks. Paul Viola and Michael Jones [23] introduced this technique which was optimized over the years as an Open Source initiative by OpenCV library contributors based on machine learning. This technique uses many images that are either positively or negatively marked for the presence of an object that is to be detected. In our case, this object would be a face. From this, the algorithm will be able to detect new facial images that are fed into the algorithm.



Fig. 4. Facial feature detection

B. Drowsiness Detection based on CNN and Facial Landmark Detection (D2CNN-FLD)

The input to the algorithm is the facial landmark coordinates which are extracted from the images using the DLib Library as represented in the algorithm 1. The presented model is based on the CNN approach with five layers. The first four convolution layers include a Linear [24] activation function with Leaky

Algorithm 1 Real-Time Driver Drowsiness Detection Algorithm with CNN

Input: DLib Facial landmark positions and labels

Output: 2 - Class probabilities of drowsy driving

- 1: Loading the Data
- 2: Min-Max Scaling to normalize the dataset between the range 0 and 1.
- 3: Adding hidden layers and dropout:
 - a. The first convolutional layer with linear activation. The input layer consists of $67 \times 2 = 134$ nodes with a total of 100 neurons.
 - b. A LeakyReLU function allows a small gradient when the unit is not active. The alpha rate is set to 10%.
 - c. A MaxPooling function to reduce the number of parameters within the model
 - d. Dropout is set to 25.4% to prevent over-fitting.
 - e. For $j=1$ to 3 do
 - (i). Adding convolution hidden layer with a linear function. The number of neurons used is 1024.
 - (ii). A LeakyReLU function allows a small gradient when the unit is not active. The alpha rate is set to 10%.
 - (iii). A MaxPooling function to reduce the number of parameters within the model
 - (iv). Dropout is set to 20% to prevent over-fitting.
 - f. Final fully connected layer with a softmax function to get a 2 class output probabilities.
- 4: The model is trained until acceptable accuracy.

ReLU [25] function to enable a small gradient when the unit is not active.

$$f(x) = \begin{cases} x, & \text{If } x > 0; \\ 0.1x, & \text{otherwise.} \end{cases} \quad (1)$$

Furthermore, MaxPooling [26] function is used to compute the maximum value from each cluster of neurons at the prior layer. In order to prevent overfitting, a Dropout function is used [24] and the dropout rate is set at 25.4% after testing out a range between 20 - 30. The output layer is the output of a Softmax [26] function to get output class label probabilities. The detailed algorithm of the D2CNN-FLD is described below.

C. Android Implementation and Architecture

An integral part of the whole process is the application that is used to capture the image data and send it for processing. The Android mobile application is capable of taking pictures of the driver. This image is analyzed by the Dlib library. Then, the Java Native Interface (JNI) [27] transmits the image data from the native Android application which is programmed in Java and the Dlib library which is developed in C++.

As illustrated in figure 5, the Dlib library on receiving the image data, will do the pre-processing and extract the facial landmarks and send the data to the above developed CNN model. This data runs through the neural network and the algorithm evaluates if the driver is drowsy or not. The results of the images are indicated on the application in a real-time fashion; if the driver is found to be drowsy, the application will send notification via visual and audio messages.

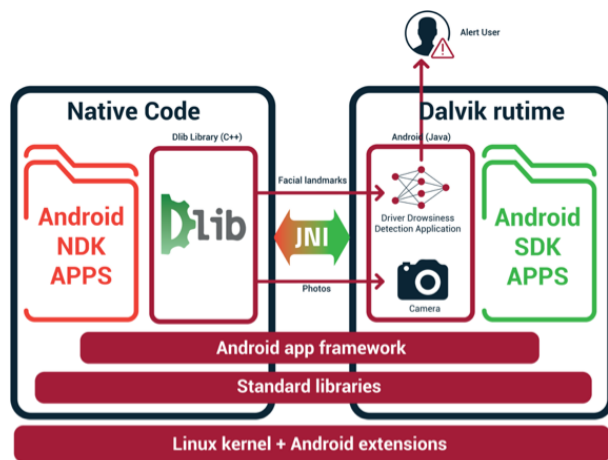


Fig. 5. Proposed architecture of solution in Android.

IV. EXPERIMENTAL RESULTS

As detailed in Table I, a total of 18 subjects having sleepy and non-sleepy status for each subject were selected for each scenario from the dataset to form the training dataset. In addition, 4 subjects were included for the evaluation. The dataset consists of 5 different simulated driving scenarios namely with glasses, with sunglasses, without glasses, night with glasses, and night without glasses. In total, 200 videos (30fps) were used and converted to images. Codebox Algorithm is used to generate new images based on the NTHU images. Furthermore, the coordinates of the facial landmark were extracted from the generated images. The number of images and videos used for each state is shown in Table I.

The experiments were conducted on a Dell workstation with Intel Xeon Gold 6128, 3.4 GHz, 64 GB RAM, and NVIDIA QUADRO P4000. The accuracy results per driving scenarios using the new technique D2CNN-FLD method are shown in Table III; its overall accuracy was 83.3%. This is a marked 2.5% improvement in the results when compared to the MLP method that was utilized in the prior study.

Evaluation and experimentation convey clearly that the eyes are a significant factor for drowsiness classification in any circumstance. In images where the eyes are obstructed by wearing sunglasses, the classification accuracy drops a few percentages. Another factor that was found to affect the classification accuracy is the illuminance on the subject's face which when increased tend to display the features on the face much more clearly. Images that lack clarity due to uneven lighting or darkness is an important performance factor since the error rate increases by 3% using D2CNN-FLD.

The main concept and contribution behind this work is to create models that are low in calculations capacity requirement so that it can be utilized in embedded systems. The experimental results show that the accuracy averages at 83.33% using the D2CNN-FLD method while maintaining a very small model size.

Experiments were performed using the model D2CNN-FLD, D2MLP-FLD [19], Faster RCNN [28] with VGG-16 [29] and

TABLE I
IMAGE CATEGORIES

Dataset	Category	Nbr. Videos	Nbr. Images Extracted	Nbr. Images detected by Dlib	Nbr. Images used After Data Augmentation
Training	With glasses	36	106,882	93,301	559,806
	Night Without glasses	36	52,372	43,659	261,954
	Night With glasses	36	50,991	39,959	239,754
	Without glasses	36	108,380	96,037	576,222
	With sunglasses	36	107,990	83,716	502,296
Evaluation	With glasses	4	37,357	34,879	209,274
	Night Without glasses	4	29,781	27,322	163,932
	Night With glasses	4	32,922	31,533	189,198
	Without glasses	4	45,005	40,714	244,284
	With sunglasses	4	28,214	25,479	152,874
Total		200	599,894	512,308	3,073,848

TABLE II
SUMMARY OF EXPERIMENTAL RESULTS. (- MEANS NOT APPLICABLE)

Dataset	Category	D2MLP-FLD MLP-Model	D2CNN-FLD CNN Model	Faster RCNN (VGG-16)	Faster RCNN (AlexNet)
Accuracy (%)	Validation	80.9	83.3	90.5	82.8
Compression (MB)	Model size	0.1	0.075	547	236
	GPU Memory	44.1	38.9	3183	845
Drowsiness Detection time (ms)	Dell workstation NVIDIA Quadro P4000	41.6	38.45	-	-
	Samsung Galaxy S8 plus	154	142	-	-
Overall speed (fps)	Dell workstation NVIDIA Quadro P4000	7.4	6.87	9.1 (GTX-1080)	22 (GTX-1080)
	Samsung Galaxy S8 plus	267.56	234.25	-	-

TABLE III
ACCURACY PER DRIVING USING CNN METHOD

Category	Accuracy(%)
With glasses	83.76
Night Without glasses	85.82
Night With glasses	79.45
Without glasses	88.89
With sunglasses	78.72
All	83.33

AlexNet [30] models are used as benchmarks for comparative purposes. A summary of the overall experimental results is shown in Table II. The models were compared in terms of accuracy, compression, drowsiness detection time, and the overall speed. The D2CNN-FLD outperformed D2MLP-FLD in both validation and test categories of accuracy. Regarding the size of the model, D2CNN-FLD is 0.075 MB that is 25% improvement over the older D2MLP-FLD which is about 0.1 MB.

Similarly, D2MLP-FLD required greater GPU computational memory (44.1MB) more than D2MLP-CNN. The two models also differ in drowsiness detection time (ms). When NVIDIA Quadro P4000 is used, in terms of time, D2MLP-FLD achieved 41.6 ms while D2CNN-FLD achieved 38.45 ms. The results are different for Samsung Galaxy S8 plus, D2MLP-FLD and D2CNN-FLD achieved, respectively, 154ms and 142ms. Moreover, the MLP achieved an overall speed of 7.4fps with NVIDIA Quadro P4000 and 267.56 fps for Samsung Galaxy S8 plus, whereas D2CNN-FLD achieved 6.87 fps and 234.25fps,

respectively.

As for model size, complexity and storage, there is a marked reduction in the new proposed model D2CNN-FLD in comparison to the older D2MLP-FLD model. As pointed out, the maximum size of the developed D2CNN-FLD model is 75 KB. On the contrary, the state-of-the-art VGG16 Based RCNN13 model comes up to 547 MB of disk space, these models require much higher RAM and computational resources. AlexNet based faster-RCNN models are equal to 236 MB, which is significantly much larger in terms of storage capacity. Although there have been improvements made in compressing these models, they still take up approximately 12 MB [31].

Furthermore, Both D2CNN-FLD and D2MLP-FLD require only 44.1 MB and 38.9 MB of RAM requirement during computation while faster-RCNN need more than 3180 MB of RAM. Hence, owing to its small model size and computing memory needs, it is possible to integrate the developed models in smaller embedded systems.

V. CONCLUSION

The paper described an improved drowsiness detection system based on CNN-based Machine Learning. The main objective is to render a system that is lightweight to be implemented in embedded systems while maintaining and achieving high performance. The system was able to detect facial landmarks from images captured on a mobile device and pass it to a CNN-based trained Deep Learning model to detect drowsy driving behavior. The achievement here was the production of a deep learning model that is small in size but relatively high in accuracy. The model that is presented here has achieved an

average of 83.33% of accuracy for all categories where the maximum size of the model did not exceed 75KB. This system can be integrated easily into dashboards in the next generation of cars to support advanced driver-assistance programs or even a mobile device to provide intervention when drivers are sleepy. There are limitations to this technology, such as obstructing the view of facial features by wearing sunglasses and bad lighting conditions. However, given the current state, there is still room for performance improvement and better facial feature detection even in bad lighting conditions.

ACKNOWLEDGMENT

This publication was made possible by an NPRP award [NPRP8-910-2-387] from the Qatar National Research Fund (a member of Qatar Foundation). The statements made herein are solely the responsibility of the authors.

REFERENCES

- [1] N. C. for Statistics and Analysis, "Crash Stats: Drowsy Driving 2015," October 2017. [Online]. Available: <https://crashstats.nhtsa.dot.gov/Api/Public/ViewPublication/812446>
- [2] M. Walker, *Why We Sleep: Unlocking the Power of Sleep and Dreams*. Scribner, 2017. [Online]. Available: <https://books.google.com.qa/books?id=8bSuDgAAQBAJ>
- [3] W. H. Organization, "The top 10 causes of death." [Online]. Available: <https://www.who.int/news-room/fact-sheets/detail/the-top-10-causes-of-death>
- [4] M. Benz, "Mercedes benz safety - s class." [Online]. Available: <https://www.mercedes-benz.co.uk/passengercars/mercedes-benz-cars/models/s-class/saloon-w222/safety/intelligent-drive.module.html>
- [5] Muze, "Eyesight." [Online]. Available: <https://www.eyesight-tech.com/>
- [6] A. D. McDonald, J. D. Lee, C. Schwarz, and T. L. Brown, "A contextual and temporal algorithm for driver drowsiness detection," *Accident Analysis & Prevention*, vol. 113, pp. 25–37, Apr. 2018. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0001457518300058>
- [7] U. S. F. M. C. S. A. T. Division, "PERCLOS: A Valid Psychophysiological Measure of Alertness As Assessed by Psychomotor Vigilance," October 1998.
- [8] C. S. Wei, Y. T. Wang, C. T. Lin, and T. P. Jung, "Toward Drowsiness Detection Using Non-hair-Bearing EEG-Based Brain-Computer Interfaces," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 2018.
- [9] V. J. Kartsch, S. Benatti, P. D. Schiavone, D. Rossi, and L. Benini, "A sensor fusion approach for drowsiness detection in wearable ultra-low-power systems," *Information Fusion*, vol. 43, pp. 66–76, Sep. 2018. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1566253517306942>
- [10] S. Tateno, X. Guan, R. Cao, and Z. Qu, "Development of Drowsiness Detection System Based on Respiration Changes Using Heart Rate Monitoring," in *2018 57th Annual Conference of the Society of Instrument and Control Engineers of Japan, SICE 2018*, 2018, pp. 1664–1669.
- [11] A. Kamilaris and F. X. Prenafeta-Bold, "Deep learning in agriculture: A survey," *Computers and Electronics in Agriculture*, vol. 147, pp. 70–90, 2018. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0168169917308803>
- [12] M. Tayab Khan, H. Anwar, F. Ullah, A. Ur Rehman, R. Ullah, A. Iqbal, B.-H. Lee, and K. S. Kwak, "Smart Real-Time Video Surveillance Platform for Drowsiness Detection Based on Eyelid Closure," *Wireless Communications and Mobile Computing*, vol. 2019, pp. 1–9, 2019.
- [13] M. F. Shakeel, N. A. Bajwa, A. M. Anwaar, A. Sohail, A. Khan, and H. ur Rashid, "Detecting Driver Drowsiness in Real Time Through Deep Learning Based Object Detection," 2019, pp. 283–296. [Online]. Available: http://link.springer.com/10.1007/978-3-030-20521-8_{_}24
- [14] L. Celona, L. Mammana, S. Bianco, and R. Schettini, "A multi-task CNN framework for driver face monitoring," *IEEE International Conference on Consumer Electronics - Berlin, ICCE-Berlin*, vol. 2018-Septe, pp. 1–4, 2018.
- [15] C.-H. Weng, Y.-H. Lai, and S.-H. Lai, "Driver drowsiness detection via a hierarchical temporal deep belief network," in *Asian Conference on Computer Vision*. Springer, 2016, pp. 117–133.
- [16] Y. Xie, K. Chen, and Y. L. Murphey, "Real-time and Robust Driver Yawning Detection with Deep Neural Networks," *Proceedings of the 2018 IEEE Symposium Series on Computational Intelligence, SSCI 2018*, pp. 532–538, 2019.
- [17] S. Mehta, S. Dadhich, S. Gumber, and A. Jadhav Bhatt, "Real-Time Driver Drowsiness Detection System Using Eye Aspect Ratio and Eye Closure Ratio," *SSRN Electronic Journal*, pp. 1333–1339, 2019.
- [18] P. Peyrard, "Personal system for the detection of a risky situation and alert," Feb. 28 2019, uS Patent App. 16/178,365.
- [19] R. Jabbar, K. Al-Khalifa, M. Kharbeche, W. Alhajyaseen, M. Jafari, and S. Jiang, "Real-time driver drowsiness detection for android application using deep neural networks techniques," *Procedia computer science*, vol. 130, pp. 400–407, 2018.
- [20] R. Jabbar, K. Al-Khalifa, M. Kharbeche, W. Alhajyaseen, M. Jafari, and S. Jiang, "Applied internet of things iot: Car monitoring system for modeling of road safety and traffic system in the state of qatar," *Qatar Foundation Annual Research Conference Proceedings*, vol. 3, p. ICTPP1072, 2018.
- [21] Codebox, "Codebox image augmentor: Data augmentation tool for images." [Online]. Available: https://github.com/codebox/image_augmentor
- [22] "dlib C++ Library." [Online]. Available: <http://dlib.net/>
- [23] P. Viola and M. Jones, "Rapid Object Detection using a Boosted Cascade of Simple Features," Tech. Rep., 2001.
- [24] "Multilayer Perceptron DeepLearning 0.1 documentation." [Online]. Available: <http://deeplearning.net/tutorial/mlp.html>
- [25] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," in *Proc. icml*, vol. 30, no. 1, 2013, p. 3.
- [26] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, <http://www.deeplearningbook.org>.
- [27] "JNI APIs and Developer Guides." [Online]. Available: <https://docs.oracle.com/javase/8/docs/technotes/guides/jni/>
- [28] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in Neural Information Processing Systems 28*, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, Eds. Curran Associates, Inc., 2015, pp. 91–99.
- [29] K. Simonyan and A. Zisserman, "VERY DEEP CONVOLUTIONAL NETWORKS FOR LARGE-SCALE IMAGE RECOGNITION," Tech. Rep., 2015. [Online]. Available: <http://www.robots.ox.ac.uk/>
- [30] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," Tech. Rep., 2012. [Online]. Available: <http://code.google.com/p/cuda-convnet/>
- [31] S. Han, H. Mao, and W. J. Dally, "Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding," 2015.