

Detection and Prediction of Driver Drowsiness for the Prevention of Road Accidents Using Deep Neural Networks Techniques



Ismail Nasri, Mohammed Karrouchi, Hajar Snoussi, Kamal Kassmi, and Abdelhafid Messaoudi

Abstract Driver drowsiness is one of the reasons for a large number of road accidents in the world. In this paper, we have proposed an approach for the detection and prediction of the driver's drowsiness based on his facial features. This approach is based on deep learning techniques using convolutional neural networks CNN, with Transfer learning and Training from Scratch, to train a CNN model. A comparison between the two methods based on model size, accuracy and training time has also been made. The proposed algorithm uses the cascade object detector (Viola-Jones algorithm) for detecting and extracting the driver's face from images, the images extracted from the videos of the Real-Life Drowsiness Dataset RLDD will act as the dataset for training and testing the CNN model. The extracted model can achieve an accuracy of more than 96% and can be saved as a file and used to classify images as driver Drowsy or Non-Drowsy with the predicted label and probabilities for each class.

Keywords Driver drowsiness detection · Deep learning · Convolutional neural networks · Transfer learning · Training from scratch

1 Introduction

Drowsiness is one of the main causes that lead to painful road accidents that take the lives of many road users in the United States. It is confirmed by statistics that 1 of 25 drivers in the age of 18 or older, had fallen asleep during the past 30 days [1, 2]. In 2013, a report was issued by the National Highway Traffic Safety Administration NHTSA, which states that drowsiness was the cause of 72,000 crashes, 44,000 injuries, and 800 deaths [3, 4]. In the most recent research conducted by the Moroccan National Highway Traffic Company [5], in 2012, on a sample consisting of about a thousand drivers, its results showed that about one out of three drivers admitted that they fell asleep while driving at least once during the month preceding the research.

I. Nasri (✉) · M. Karrouchi · H. Snoussi · K. Kassmi · A. Messaoudi
Electrical Engineering and Maintenance Laboratory, High School of Technology, Mohammed
First University, BP. 473, Oujda, Morocco
e-mail: i.nasri@ump.ac.ma

© Springer Nature Singapore Pte Ltd. 2022
S. Bennani et al. (eds.), *WITS 2020*, Lecture Notes in Electrical Engineering 745,
https://doi.org/10.1007/978-981-33-6893-4_6

57

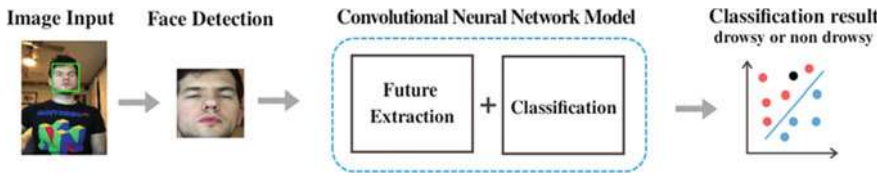


Fig. 1 Architecture of drowsiness detection system

The results also revealed that 15% of them stated that they had driven for five hours (about 500 km) without stopping, while 42% of them stated that they stopped only once during the same distance, although the normal situation is to make two stops as a minimum. This study will focus on the (UTA-RLDD) University of Texas analysis in the Arlington Real-Life Drowsiness Dataset [6]. The figure below presents the architecture of Driver Drowsiness Detection, in three phases: face detection, feature extraction and classification. If a driver's face is found, Viola-jones [7, 8] face detection algorithm is used to detect and crop the driver's face from the image and it will be given as input to CNN. The Feature Detection Layers in CNN are used to extract the deep features which will be passed to Classification Layers. Softmax layer in CNN classifies the images as drowsy or non-drowsy and gets results of predicted label and probabilities. An alert system is used when the model detects a drowsy state continuously (Fig. 1).

The rest of this paper is organized as follows. In Sect. 2, a brief description of the related work is presented. The Sect. 3 provides an overview of the proposed solution and approach to prepare a deep learning model. The results obtained from experiments are discussed in Sect. 4. Finally, we conclude in Sect. 5 with future directions.

2 Related Work

Several systems and approaches have been proposed for detecting the driver drowsiness. In this section, a review of the previous methods and approaches to detect drowsiness based on extracting facial features will be provided. Jabbar et al. [9] developed an approach based on extracting landmark coordination from images using Dlib [10] library. This approach can classify the driver's face as drowsy or non-drowsy based on his face landmark. In fact, the facial landmark detector implemented inside Dlib produces 68 (x, y) coordinates to describe specific facial structures of the face. Dlib is a general purpose platform software library written in the programming language C++ to provide a Machine Learning algorithm used in a wide range of fields and applications. Danisman et al. [11] proposed a method to detect drowsiness based on monitoring the changes in the eye blink duration. In this matter, CNN based eye detector was used to find the location of the eyes and to calculate the "no blinks" per minute. If the blink duration increases, this indicates that the driver

becomes drowsy. In this study, we take into consideration all the signs that show that the driver is drowsy (eye color and shape, yawn, and blink). All these signs are related to the face of the driver. For this purpose, we use the cascade object detector that uses the Viola-Jones algorithm for detecting and extracting the driver's face from images. These extracted images will act as the dataset for training and testing the Convolutional Neural Networks CNN proposed. The Viola-Jones [7, 8] object detection, developed by Paul Viola and Michael Jones in 2001, is the most popular object detection algorithm to provide competitive object detection rates in real-time. It can be used to solve a variety of detection problems, including the problem of face detection.

3 Proposed Solution

This section provides an overview of the proposed solution: Dataset and approach to prepare a CNN model that will be used to classify images of the driver as Drowsy or Non-drowsy.

3.1 Dataset and Preprocessing

About the dataset creation, this study will focus on the University of Texas analysis in the Arlington Real-Life Drowsiness Dataset (UTA-RLDD) [6]. It contains the full component of the dataset for training and testing. From this dataset, 28 subjects were selected from 60 subjects available. Subjects were instructed to take three videos from the phone or the webcam; in three different drowsiness states according to the KSS table [12]. In this work, we focus on two classes (see Fig. 3); these classes were explained to the participants in the following way:

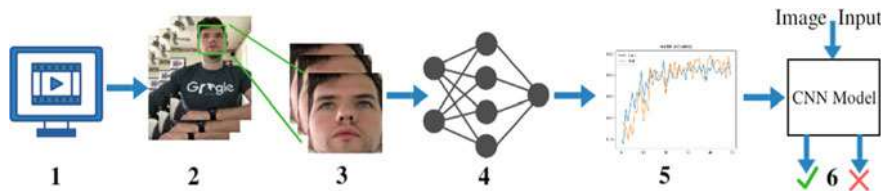
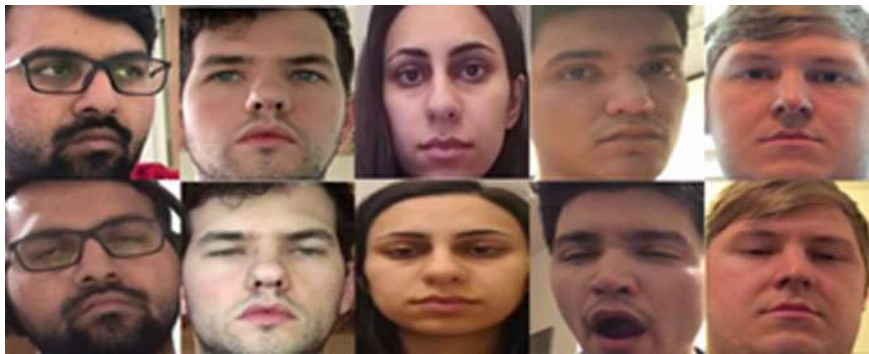
- Non-Drowsy: In this state, subjects were told that being alert meant they were completely conscious and they can drive easily for long hours [6], as illustrated in level 1, 2 and 3 in the KSS table [12].
- Drowsy: This condition means that the subject needs to resist falling asleep, as illustrated in level 8 and 9 in Table 1.

3.2 Proposed Approach

In this section, an overview of the proposed approach (see Fig. 2) to prepare a CNN model will be provided. The proposed approach consists of six main steps:

Table 1 KSS table [12]

Rating	Verbal descriptions	Rating	Verbal descriptions
1	Extremely alert	6	Some signs of sleepiness
2	Very alert	7	Sleepy, but no difficulty remaining awake
3	Alert	8	Sleepy, some effort to keep alert
4	Rather alert	9	Extremely Sleepy, fighting sleep
5	Neither alert or sleepy		

**Fig. 2** Approach proposed to prepare a CNN model**Fig. 3** Non drowsy and drowsy image samples detected by the Viola-Jones algorithm from the dataset

- Step 1: Selecting videos from RLDD Dataset: The videos were selected from the Real-Life Drowsiness Dataset RLDD based on a variety of simulated driving scenarios and conditions.
- Step 2: Extracting Images from selected videos: The frames were extracted from videos as images using VLC software.
- Step 3: Detecting and Cropping the driver's face from images: In the third step, we use the cascade object detector that uses the Viola-Jones algorithm to detect and crop the driver's face from images (see Fig. 3). These images will be used for training and testing the proposed models (70% for training and 30% for testing).
- Step 4: Creating and Configuring Network Layers: In this step, we define the convolutional neural networks CNN [13] architecture.

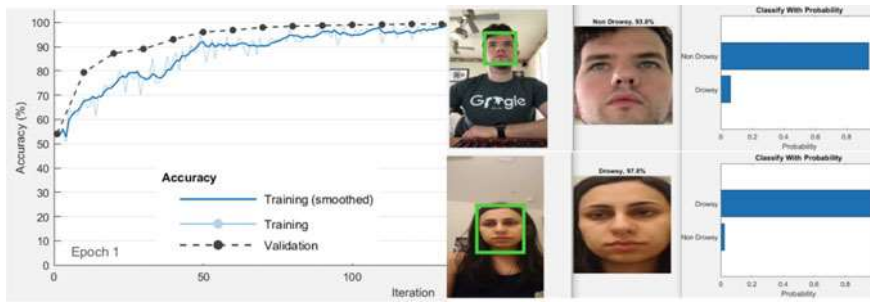


Fig. 4 The training progress plot for transfer learning and Classification images with the predicted label and the predicted probabilities %

- Step 5: Training and testing the model: The cropped driver's face will act as the input for the algorithm detailed in (algorithm 1 and 2). The model uses Deep Neural Networks Techniques and was trained using 2 methods: Training via Transfer learning (algorithm 1) and Training from Scratch (algorithm 2).
- Step 6: Extracting the model: Finally, the CNN model can be saved as a file and used to classify images with the predicted label and probabilities (see Fig. 4).

3.3 Training via Transfer Learning

For the transfer learning, we use AlexNet [14] to classify the images by the extracted features. AlexNet is a CNN that contains eight layers and can classify images into 1000 object categories, such as a laptop, pen and many objects. In order to make the AlexNet recognize just two classes, we need to modify it. The network was trained by the following algorithm.

Algorithm 1: Training via Transfer learning

Input: Driver's face dataset and labels

Output: Learned CNN model

1. Load and Explore Image data from My PC (Driver Face)
 2. Specify Training and Testing Sets (Split data into training and test sets)
 3. Load Pre-trained Network (AlexNet)
 4. Modify Pre-trained Network (AlexNet): We modify final layers to recognize just 2 classes (drowsy and Non-drowsy)
 5. Specify Training Options
 6. Train New Network Using Training Data
 7. Classify Test Images and Compute accuracy (see Fig. 4)
-

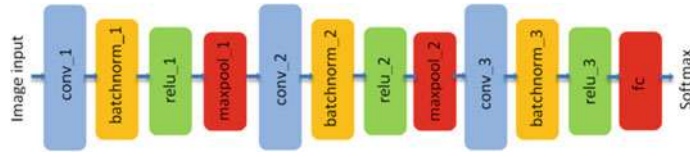


Fig. 5 Proposed deep CNN model

3.4 Training from Scratch

For the training from scratch, we are creating and configuring network layers by defining the convolutional neural network architecture and training the network by the following algorithm.

Algorithm 2: Training from Scratch

Input: Driver's face dataset and labels

Output: Learned CNN model

1. Load and Explore Image data from My PC (Driver Face)
 2. Specify Training and Testing Sets (Split data into training and test sets)
 3. Create and Configure Network Layers by defining the convolutional neural network architecture. In the proposed CNN model, we use 3 convolutional layers and one fully connected layer. Softmax classifier is used to classify images as drowsy or non-drowsy
 4. Specify Training Options
 5. Train Network Using Training Data (imdsTrain)
 6. Review Network Architecture (see Fig. 5)
 7. Classify Test Images and Compute accuracy
-

4 Experimental Results

In this section, we will present the results of the training CNN models by two commonly used approaches for deep learning: transfer learning and training from scratch. In this work, 28 subjects were selected from 60 subjects available in the Arlington Real-Life Drowsiness Dataset (UTA-RLDD) [6] to obtain training and testing data. For data processing, the frames were extracted from videos as images using VLC software. After that, the driver's face was detected and cropped from images using the Viola-Jones algorithm as it appears in the Table 2.

The processor for training and test processing platform was a 3.6 GHz Intel (R) Core (TM) i5-8350U with 8 GB memory and 256 GB SSD hard disk. The development platform for the algorithm was MATLAB R2018b.

In the rest of this section, a comparison of training the model from scratch and transfer learning is presented. In this paper, the two models were trained and evaluated by the same number of images dataset (101,793 images). Table 3 shows the network performance of these models. Training the model from scratch and achieving

Table 2 Overview of images extracted and detected by Viola-Jones algorithm

Dataset	Nbr of extracted images	Nbr of images detected by the Viola-Jones algorithm
Drowsy	55,600	52,348
Non-Drowsy	53,586	49,445
Total	109,186	101,793

Table 3 Comparison between the proposed models (CNN transfer and CNN scratch)

Model	Size (Mbit)	Accuracy (%)	Training time	Training and testing Images
CNN transfer	622	93	128 min 54 s	101,793
CNN scratch	2.38	96	159 min 11 s	101,793

reasonable results requires a lot of effort and computer time, which is due to the time needed to test the performance of the network; if it is not adequate, we should try modifying the CNN architecture and adjusting some of the training options and then retraining. The training time of the CNN Scratch model is 159 min and 11 s. Experimental results show that the accuracy rate of the developed model is almost 96% for training from scratch and 93% for transfer learning. Training the model with transfer learning is much faster and easier, and it is possible to achieve higher model accuracy in a shorter time (higher start) but with large model size. The maximum size of the developed models is equal to 622 Mbit for CNN Transfer and 2.38 Mbit for CNN Scratch.

5 Conclusion

In this work, we have proposed a method for driver drowsiness detection based on his facial features. The face is detected using the Viola-Jones algorithm. The proposed CNN with Feature Detection Layers is used to extract the deep features, and those features are passed to Classification Layers. A Softmax layer in the CNN provides the classification output as driver drowsy or non-drowsy and the probabilities for each class. The proposed model has been trained and evaluated using Real-Life Drowsiness Dataset (RLDD) by two commonly used approaches for deep learning: transfer learning and training from scratch. On the one hand, the results show that the size of the proposed model for training from scratch is small while having an accuracy rate of 96% but with a lot of effort and computer time. On the other hand, with transfer learning, we can achieve an accuracy of 93% with less computer time and effort but with large model size. Further work will focus on the implementation of the model in an embedded system and the creation of an integrated alert system into the vehicle to wake the driver up before anything undesired happens.

References

1. Drowsy driving 19 states and the District of Columbia (2009–2010). Retrieved from <https://www.cdc.gov/mmwr/pdf/wk/mm6151.pdf>
2. Drowsy driving and risk behaviors 10 states and Puerto Rico (2011–2012). Retrieved from <https://www.cdc.gov/mmwr/pdf/wk/mm6326.pdf>
3. National Highway Traffic Safety Administration. Research on Drowsy Driving external icon (October 20, 2015). Retrieved from <https://one.nhtsa.gov/Driving-Safety/Drowsy-Driving/Research-on-Drowsy-Driving>
4. The Impact of Driver Inattention on Near-Crash/Crash Risk (April 2006). Retrieved from <https://www.nhtsa.gov/>
5. lakome2. <https://lakome2.com/relation-publique/119885>, (30 May 2019)
6. Ghoddoosian R, Galib M, Athitsos V (2019) A realistic dataset and baseline temporal model for early drowsiness detection. In: The IEEE conference on computer vision and pattern recognition workshops
7. Viola P, Jones M (2001) Rapid object detection using a boosted cascade of simple features. In: The 2001 IEEE computer society conference on computer vision and pattern recognition. CVPR 2001, vol 1. IEEE
8. Viola P, Jones M (2001) Robust real-time object detection. *Int J Comput Vis* 4(34–47):4
9. Jabbar R, Al-Khalifa K, Kharbeche M, Alhajyaseen W, Jafari M, Jiang S (2018) Real-time driver drowsiness detection for android application using deep neural networks techniques. *Procedia Comput Sci* 130:400–407
10. Dlib C++ toolkit. Retrieved from <https://dlib.net/> (2018 Jan 08)
11. Danisman T, Bilasco IM, Djeraba C, Ihaddadene N (2010) Drowsy driver detection system using eye blink patterns. In: 2010 international conference on machine and web intelligence. IEEE, pp 230–233
12. Åkerstedt T, Mats G (1990) Subjective and objective sleepiness in the active individual. *Int J Neurosci* 52(1–2):29–37
13. O'Shea K, Nash R (2015) An introduction to convolutional neural networks. arXiv preprint [arXiv:1511.08458](https://arxiv.org/abs/1511.08458)
14. Krizhevsky A, Sutskever I, Hinton GE (2012) Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems