Communications in Mathematics and Applications
Vol. 13, No. 2, pp. 807–822, 2022
ISSN 0975-8607 (online); 0976-5905 (print)
Published by RGN Publications
DOI: 10.26713/cma.v13i2.2034

http://www.rgnpublications.com

Review Article

Real-Time Driver Drowsiness Detection System Using Dlib based on Driver Eye/Mouth Monitoring Technology

Wesam H. Al-Sabban

Department of Information Systems, College of Computer and Information Systems, Umm Al-Qura University, Makkah, Saudi Arabia
whsabban@uqu.edu.sa

Received: December 19, 2021 **Accepted:** February 20, 2022

Abstract. The driver sleepiness is common and one of the main causes of road accidents. So, there is a need for automatically detecting of this human behaviour. In any case, if the drivers feel drowsy, they still keep on driving the vehicle, and accidents occur. This study can be implemented using the CNN training model and initiating an alarm if the drowsiness condition is detected. Many of the authors suggested the process of detecting the Drowsiness (Problem Statement) of the drivers using technologies like the Internet of Things (IoT), Deep learning, and Haar cascade (to detect the coordinates of eyes and mouth, which are the target objects). However, this study contributes towards providing the real-time application in co-operating the CNN model and Dlib. Hence, this study proposes a novel embedded system with CNN technology. The CNN model is fed with inputs based on four (4) images related to eye and mouth openings and closings. This application is trained using the CNN model, which takes inputs as images and processes by identifying the features on the face using the Dlib library while representing the change in the state of coordinates of eyes and mouth as Yawning. This approach is achieved using Convolution Neural Network (CNN), pillow, Pygame, OpenCV, and the Dlib, along with providing an alarm when the position of mouth changes. The model is recorded with a maximum validation accuracy of 98% with the minimum recorded loss of less than 0.04% as areal-time application.

Keywords. Drowsiness, CNN, IoT, Dlib

Mathematics Subject Classification (2020). 68T05, 68T07, 68T45

1. Introduction

1.1 Causalities of Accidents on Roads

According to World Health Organisation (WHO) estimates from 2015, road traffic injuries resulted in around 1.25 million deaths globally, approximately every 25 seconds [29]. While the projected cost of road accidents in Europe is over 160 billion Euros, according to the Americans accident national highway traffic safety administration, driver sleepiness is responsible for around 100,000 accidents every year in the United States alone (NHTSA) as shown in Figure 1.

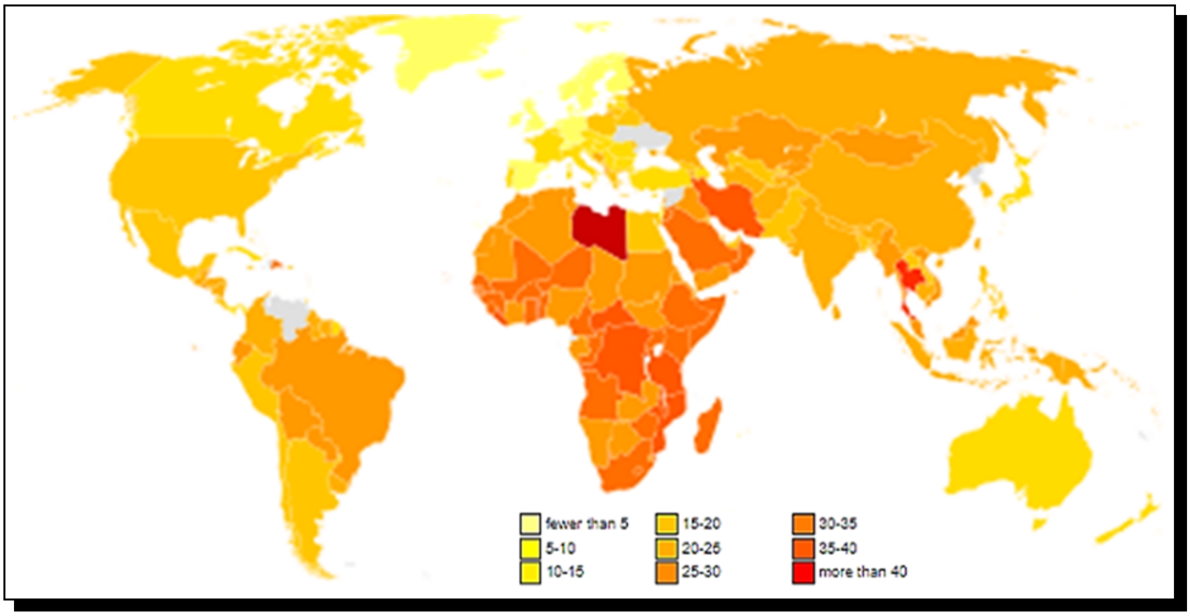

Figure 1. Death rates by accidents per country[1]

There are more than 25% of road accidents occurring according to the World Health Organization (WHO) [29]. However, biologically humans tend to underestimate the causes of dangerous sleep that results in fatal accidents. Numbers representing the overall deaths and accidents might increase or maybe higher [26]. However, the crashes due to Drowsiness seem fatal similar to that of alcohol indulged accidents. This is quite sensible that no sleep for 18 hours is similar to that of contents of alcohol in blood, i.e., BAC of 0.05% [1]. When a person reaches this stage, he/she is likely to lose concentration.

However, one in 25 drivers accepts that they doze off while driving for long hours, and more than 25% of them have noticed Drowsiness while driving, which is noticeably dangerous [12]. These incidents often happen due to the physical/reasons of the driver. However, there might be many other contributing factors that may cause it scientifically.

However, from the map and considering most of the inhabitants from the country, it can be deduced that more than 40 per cent of death rates is the highest [20]. However, this is considered as an overall death due to road accidents, not considering the deaths caused to just the Drowsiness. One of the main effects caused by feeling sleepy while driving is the inability to

[1]Source: https://en.wikipedia.org/wiki/List_of_countries_by_traffic-related_death_rate

focus correctly [25], which is one of the main reasons behind the delay, which causes to misjudge the distances [7]. Hence, at these stages, the body undergoes a transition. Therefore, the level of this Drowsiness depends on various other factors.

Table 1. Report on leading factors which causes fatal Crashing's

Factors influencing the accidents	Percentage of occurrences
Driving faster	16%
Confusion due to Vision	3%
Distractions like using mobile	5.2%
Due to Drowsiness	1.2%

1.2 Levels of Driver's Drowsiness

There are some levels of Drowsiness a person can experience, which is somehow related to psychology, measures on vehicles, and sometimes behavioural [27]. Most of the conditions associated with the psychology of the drivers are gathered by adding the devices (electronic). This type includes the components like Electroencephalography (EEG) [6]. These components give out high accurate results. Other physical movements like wheel movements patterns in brake and departure movements are considered for the second level of Drowsiness. However, the measurements from the steering wheels are widely used to gain better results. The other type of measurement is based on the person's behaviour making it more reliable than the vehicle-based. The information is obtained from the cameras to detect the movements like facial expressions. This has become one of the popular methods as it explains more about the outcomes [13].

2. Generalised Process of Drowsiness Detection

There are specific steps involved in detecting the driver's Drowsiness [16], they are:

2.1 Generic View of Required Hardware

Lachat *et al.* used a camera and placed it in front of the user. Pictures were relayed to a processor using this manner by [16]. The identification of facial characteristics is a restructuring process of templates. This technique entails comparing facial expressions, particularly eyes and photographs. Facial features expression reorganisation by use of Viola-Jones detection. In 2001, Power Viola and Michel Jones developed the Viola-Jones algorithm. This technique is commonly used to identify eyeballs but is typically used to detect objects [14]. This method requires a significant amount of time to train. The detection procedure, on the other hand, is quick and straightforward. This technique is used in real-time objects detection. Some algorithms are used in a cascade order to facilitate and accelerate the detection process [30].

2.2 Find Eyes on the Face

Many approaches have been developed to locate the position of the eyes. In many circumstances, the face of the driver is used to identify eyes. One simple method is to use the characteristics of the eyes. People have different colour spectrums in their eyes depending on the type of habitat they grow [28]. In this work, blink patterns and yawning patterns are used. RGB format is sometimes not considered ideal for this kind of approach since it is used to establish the colour code of one pixel. Other parameters such as lighting and saturations are crucial in addition to hue. Therefore, Other formats, like lighting, saturation, and colour, are considered.

2.3 Feature Extraction

Feature extraction is a proportionate decreasing procedure that reduces a large amount of raw data into smaller groupings for processing [5]. One of the features of the huge datasets is that a more significant number of variables need a lot of computational resources to process. Feature extraction is beneficial when researchers need to decrease the number of resources required for processing without losing vital or relevant information. The feature can also help to reduce the quantity of duplicated data for a specific study. Furthermore, the reduction of data and the computer's efforts in creating variables combination (features) speed up the learning and generalisation processes in the machine learning process [17].

Feature extractions real-world applications are Autoencoders, bag-of-words, and image processing. Where Autoencoders are designed to learn different data coding, which is unsupervised. The feature extraction technique is utilised to find important characteristics in the data for coding the original data set to derive new ones. This method may also be used to process images. In image processing, algorithms are used to recognise characteristics of videos such as shapes, edges, or movements. To begin, a user must understand how a machine reads and stores pictures. Loading the images, reading them, then processing them through the computer is tough since the machine does not have eyes like people. Machines see any image as a numerical matrix. The number of pixels in the incoming image determines the size of this matrix [9].

Each pixel value explains how bright that pixel is and it's colour. In the simplest example of a binary picture, the pixel value is a 1-bit integer representing both foregrounds. Hence, pixels are the numbers that indicate the intensity. Smaller values closer to zero represents black, whereas bigger ones closer to 255 make it white [18]. So, this is the nature of pixels and how machines see images without eyes by using numbers. The actual image is 28×28 inches. And if the user wants to verify it, they can do so by counting the number of pixels.

However, in the case of a coloured image, researchers have three matrices to work with: RED, GREEN, and BLUE. So, in these three matrices, each matrix contains a value between 0-255 that indicates the intensity of the colour of that pixel [3]. These three channels are overlaid and utilised to create a coloured picture.

Programmers transform images to binary form after detecting shut and open eyes, as well as blinks [32]. This conservation reduces the quality of data while handling systems. The picture is then separated into top and bottom halves. When the eyes are open, the pupils and eyelashes are dark; therefore, the ratio of dark pixels in the upper region to the bottom half is greater than when the eyes are closed [32]. This is because the eyelids cover the eyes during eye closure, and the eyelashes are located in the bottom region of the eye.

The image's lighting should be adjusted before binary conversion. To normalise a picture, it is turned to grayscale. Then, lighting will be proven by paying attention to the V average [24].

Following that, the picture is transformed to grayscale. The picture is then converted to binary using a threshold. To establish the threshold, they used the threshold automatically. Light pixels are threshold automatically. Light pixels are regarded as white, whereas dark pixels are considered black. Image resolution may be improved by dilation and Erosion. These two are used to eliminate tiny spots. Then, the ratio of black pixels to total pixels was computed in both the top and lower parts of the head. This ratio is used to distinguish between eyes that are closed or open. This helps in diving towards the behavioural pattern of understanding the level

2.4 Significance of CNN into Detection of Driver's Drowsiness

The basic layers of CNN are the convolutional layer, pooling layer, a ReLU layer, and a fully-connected layer. Each kernel (filter) in the convolutional layer has three things: width, depth, and height. The feature maps are generated by computing the scalar product of the kernel and local images areas. CNN uses pooling layers to reduce the size of features maps to accelerate the computations. In this layer, the input is an image separated into regions; operations are done on each area based on the regions. Max pooling selects a maximum value for each region and sets it in the corresponding position in the output [31].

Usually, a nonlinear layer is Rectified Linear Units (ReLU), the max function to all the values in the input data, and sets all negative values to zero. The ReLU activation function is depicted in equation (2.1) below:

$$f(x) = \max(0, x), \tag{2.1}$$

whereas the fully connected layer is utilised to generate class scores from activation needed for classification. The deep CNN model layer is designed to detect driver's Drowsiness using deep learning based on the Eye states. A novel deep CNN model is created in the suggested work by Zhu and Zabaras ConvolutionNetwork. In which four convolutional layers (4) and one fully connected layer (1) are employed in the suggested technique. Extracted key pictures with dimensions of 128×128 are sent into the convolution layer-1 (Conv2d 1) [34].

Conv2d _1 convolves the input picture using 84 3×3 filters. 840 parameters were required by Conv2d_1. Batch normalisation 1 has 336 parameters. The output of convolution layer-1 is passed into convolution layer-2 (Conv2d 2). Conv2d_2 convolved input with 128 5×5 filters. Following convolution, batch Normalisation, nonlinear transformation ReLU, MaxPooling across 2×2 cells with stride 2, and dropout with 0.25 per cent applied [25]. Conv2d_2 needs 268928 parameters. The output of convolution layer-2 is passed into convolution layer-3 (Conv2d 3). Conv2d_3 convolved input with 256 5×5 filters. Following convolution, Batch Normalization, and nonlinear transformation ReLU, MaxPooling over 2×2 cells with stride 2 followed by dropout with 0.25 per cent applied, Conv2d_3 required 819456 parameters. Convolution layer-4 (Conv2d 4) receives the output of convolutional layer-3 [11].

A total of 12,757,874 trainable parameters were required for the proposed CNN model because the classifier's output is two states, the output layer only has two outputs [4]. The Adam approach is used for optimisation. The SoftMax classifier is utilised for classification in this case. In this proposed CNN structure, the 256 outputs of the fully connected layer are the in-depth features extracted from the input eye pictures. The final two outputs can be linear combinations of profound characteristics.

In the suggested system for detecting driver drowsiness by Zhou *et al.* [33], a convolutional neural network (CNN) is employed. The author here explains object detection using a deep learning network. CNN is considered here because each sleepy feature-based image requires a feature vector that can be compared to existing characteristics in a database to determine whether or not it is drowsy. Typically, CNNs require fixed-size pictures as input; therefore, preprocessing is essential. Preprocessing comprises extracting important frames from video based on temporal changes and storing them in a database. CNN's convolutional layer creates features vectors from these stored pictures. These feature vectors are then utilised to identify driver sleepiness. The author elaborates on the basics of feature extraction from the faces detected as well as the face detection using Deep learning. In generic, it is to localise the face from the object detected. This makes it a specialised task [33]. However, the author expresses that one of the famous specialised processes towards Face detection is Haar Cascade and AdaBoost. No shreds of evidence were noted from Zhou *et al.* that Dlib is used to detect faces.

Similarly, Reddy *et al.* [22] explained that the state of the drivers is critical since it is one of the leading causes of motor vehicle accidents and drivers' inattention or tiredness. The technique used in sleepiness detections is based on a deep learning technology that can be deployed on Android applications. The author also stresses that a sleepiness detector in an automobile can help to prevent many accidents. Accidents occur as a result of a single moment of neglect. This device should be portable and accurate enough to be used in an embedded device. This work proposes a unique technique for real-time sleepiness detection based on deep learning that can be implemented on a low-cost integrated board and performs well. Therefore, using this technique, the accuracy obtained is 89.5%; that is, the drivers' sleepiness can be detected easily. Jabbar *et al.* [8] illustrated that road traffic and other types of accidents are a leading source of injury and death in the human population using 22 subjects in their study with different ethnicities. However, Multi-Layer Perceptron (MLP) is used to train the model, and Dlib extracts the face Landmarks. The model gained around 81 % of accuracy as the model size is considered smaller.

However, the study shows that their network can be easily imported into the embedded systems and deployed as a real-time application [22]. The subjects considered in the study also look similar to this study, whereas the study proposed by Reddy *et al.* [22] evaluated the drowsy state where the driver's face is not visible.

However, the work done by Reddy *et al.* [22] achieved 89% accuracy using a heavy baseline light-weighted model. This model is aimed towards deployment into the embedded system for real-time application. The work done here does not consider any such constraints. The constraints covered here are the position of the camera.

Hence, the present work aims towards providing better results compared to other works by Dua*et al.* [15] (which is implemented using four (4) different architecture yet the accuracy was seen as 85%) is unique. Whereas the present study records the highest prediction accuracy of 98% with a minimum loss, this work records 98% of accuracy. Haar cascade is better known among the machine learning approaches to identify the faces and their features like eyes and mouth. This includes finding and calculating the pixel intensities, which are pre-trained and hustle-free. The process trims down the unwanted parts similar to that of Dlib. But Dlib is quite powerful while detecting even the smaller faces out with less noise (more precise cropping of features). This makes Dlib quite powerful and better as an option. Hence, this study is carried out using Dlib's facial detector.

3. Methodology

Various studies suggest the use of Haar cascade in determining the Features coordinates and face detection. None of the studies includes the use of Dlib for face detection techniques [32-34]. The face features considered here in the survey are eyes and mouth coordinates.

The Detection process is carried out into three (3) steps, as seen in Figure 2. Extraction of landmarking, Model Training, and Implementation using OpenCV.

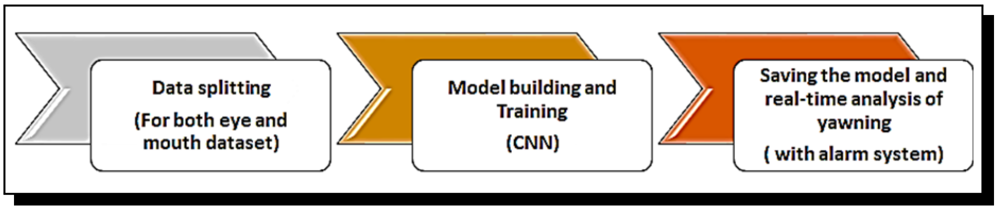

Figure 2. Process flow of detection of drowsiness using CNN

Extraction of Facial Landmarking

This is considered the first step towards data splitting and processing. The reference images in Figure 3 and Figure 4 are used to obtain the coordinates of the land markings. These features are then localised and extracted to minimise the maximum background noise. Basic features extracted are nose, eyes, mouth, eyebrows, and jawline. These facial landmarks extracted can be used in many applications like an alignment of the face, swapping of faces, e.g., Snapchat, and other applications like blinking detection.

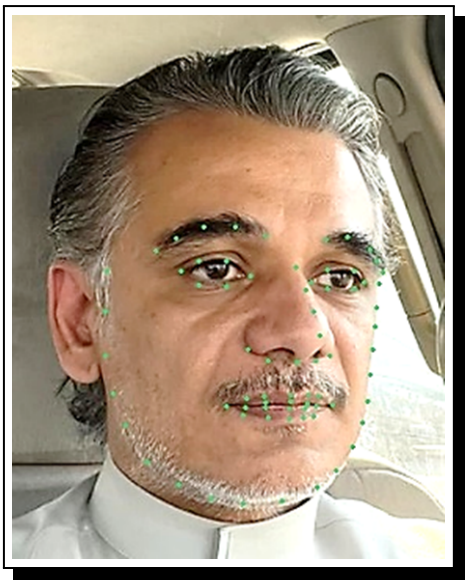

Figure 3. Representation of face landmarking

These facial landmarks extracted can be used in many applications like the alignment of the face, swapping of faces like in applications like Snapchat, and other applications like blinking detection.

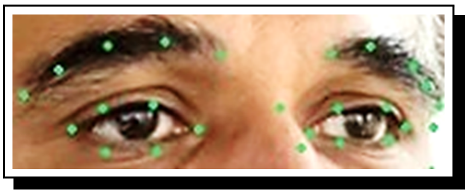

Figure 4. Representation of eye landmarking

The coordinates of the eyes are extracted from the numbers 37 to 42 for the right eye and 43 to 46 for the left eye. The positions can be seen in Figure 4 as a reference.

Figure 5. Representation of mouth landmarking

Similarly, the mouth positions are noted from 49 to 68, as shown in Figure 5. However, the formula to calculate the Eye Aspect Ratio (EAR) is calculated using the coordinate points from Figure 3. EAR is defined as the ratio of width and height of the human eye and twice the width of an eye. The average values for EAR are 0.33 and 0.141 for opening and closing of eyes [19].

CNN Model Training

After data is split and processed, the model is implemented. The images are rescaled using Image Data Generator, and a Sequential model with 495,140 trainable number parameters is implanted.

Real-time Implementation

The implemented model is then saved to make predictions after it is tested on the test data. The classification reports are generated. This protected model is then implemented using OpenCV, which facilitates the real-time video capturing the feed from the camera fixed in the car, as shown in Figure 6.

The application maps the facial landmarking (Eye/Mouth) using the Dlib and CNN model to predict whether the mouth markings changed from the default. If it does, then the warnings are given using OpenCV. This application is then deployed into the embedded system. The camera's position is considered near the driver's seat in the car at a moderate distance to cover the face of the driver.

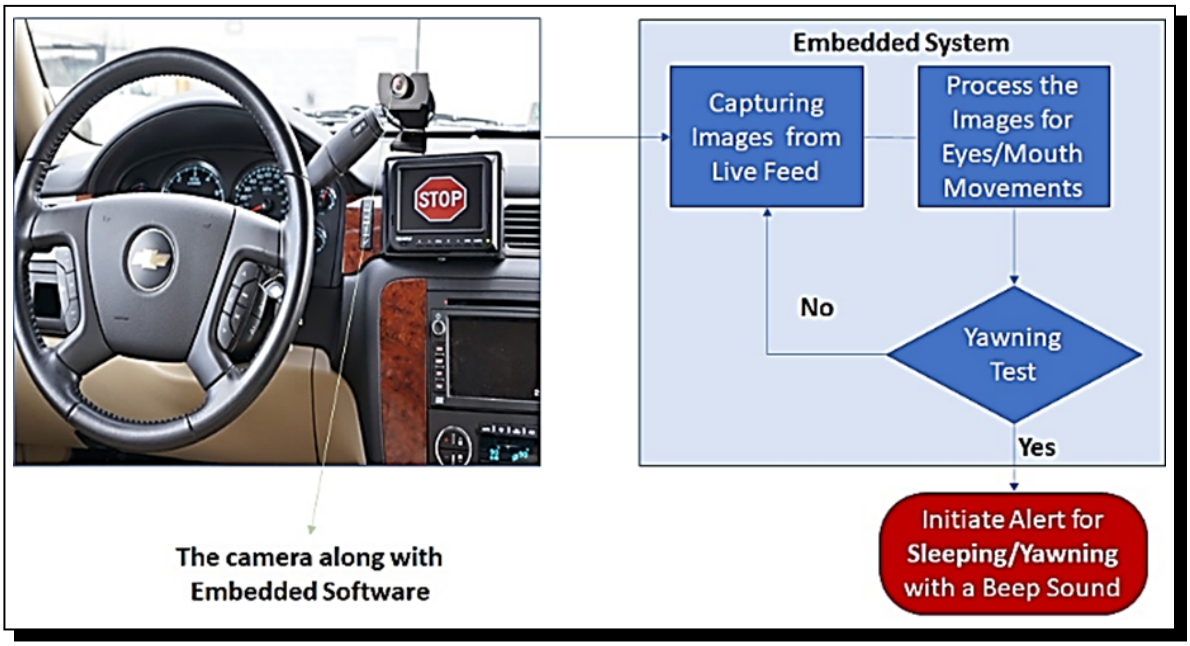

Figure 6. Process flow of detection of drowsiness using CNN

4. Results

The dataset is used as an input to the CNN model. The dataset consists of various classes like eye closed, eye-opening, mouth closing, and mouth opening. The sample images present in the dataset are shown in Figure 7.

Figure 7. Representation of dataset splitting

Data Splitting and Organisation

The dataset is split into eyes and mouth. The Dlib frontal detector is used to find the faces in the images. From the images, the mouth part of the face is cropped and padded based on the land markings from the Dlib. Similarly, these steps are repeated for finding and cropping the eyes as well from the raw dataset. These images are then converted to NumPy.

The images where the mouth part is cropped are again processed to filter out the images with "yawn" and "no yawn". The Yawn is calculated based on the distance between land markings from the mouth part from 61 to 64. Finally, the images are then saved with two different yawn values, and the formula can be seen in Figure 7. One which exceeds 0.4 and the other which is less than 0.05. Similarly, the distance between the landmark points' left and right eyes are calculated. These distances are then used as a reference to separate as eye-opening and closing.

CNN Model Implementation and Training

In this step, the segregated data is used and divided into four (4) classes that are eye closed, eye open, mouth closed, and mouth open. The images are sent to preprocessing, where the images are resized as (145,145). Images are usually converted from RGB (Red, Green, Blue) to grayscale to reduce the complexity and noise. But, here in this study, the images are kept in RGB scale to analyse the real-time issues and keep them real. After this, the data is formally shuffled. X (input) is appended with the array values, and y (output) is filled with labels. Then X is converted into NumPy arrays. Then finally, the training and testing data is split with a 20% testing size. Hence, the model is implemented, and the summary of the model is seen in Figure 8.

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 143, 143, 256)     7168
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 71, 71, 256)       0
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 69, 69, 128)       295040
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 34, 34, 128)       0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 32, 32, 64)        73792
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 16, 16, 64)        0
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 14, 14, 32)        18464
_________________________________________________________________
max_pooling2d_3 (MaxPooling2 (None, 7, 7, 32)          0
_________________________________________________________________
flatten (Flatten)            (None, 1568)              0
_________________________________________________________________
dropout (Dropout)            (None, 1568)              0
_________________________________________________________________
dense (Dense)                (None, 64)                100416
_________________________________________________________________
dense_1 (Dense)              (None, 4)                 260
=================================================================
Total params: 495,140
Trainable params: 495,140
Non-trainable params: 0
```

Figure 8. Model summary

Hence, the model is then trained at 50 epochs. The accuracy and loss variations for every epoch are shown in Figure 9.

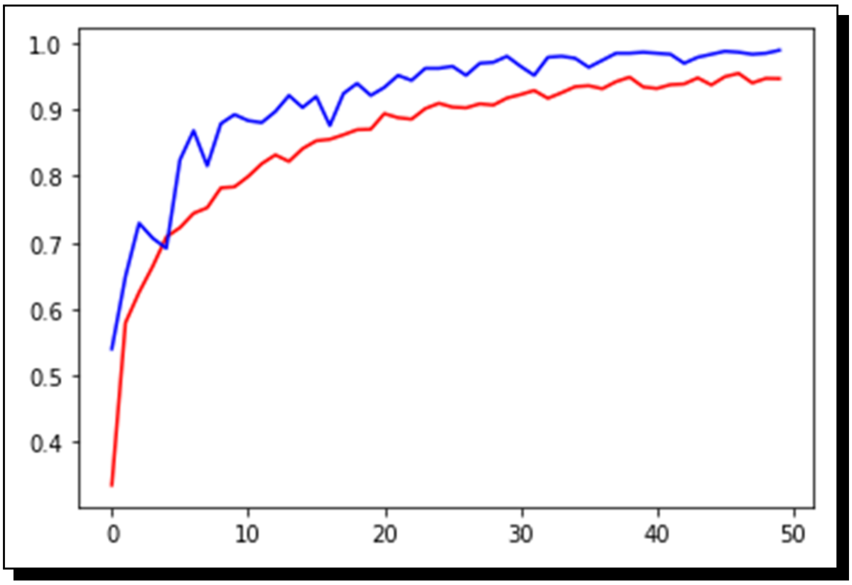

Figure 9. Model accuracy

The validation accuracy and loss are denoted with blue colour, training accuracy and loss are indicated with red colour as shown in Figure 10.

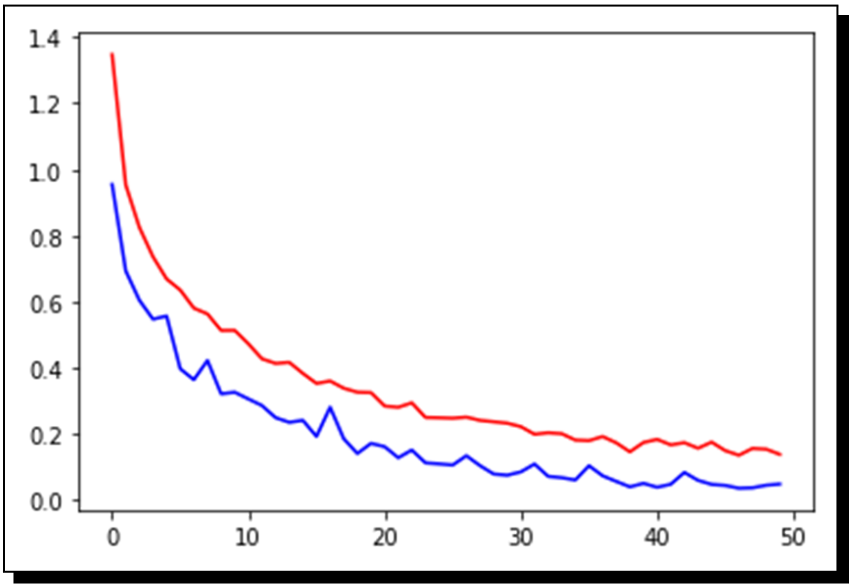

Figure 10. Model loss

The highest recorded validation accuracy is 98%, with 0.04 loss functions. The model is then saved with a JSON and YAML file format. This is then implemented in the test data set. The classification report for the model is seen in Figure 11.

	precision	recall	f1-score	support
eyeclosed	1.00	0.99	0.99	260
eyeopen	0.98	1.00	0.99	226
mouthclosed	0.96	0.99	0.97	174
mouthopen	0.99	0.95	0.97	160
accuracy			0.98	820
macro avg	0.98	0.98	0.98	820
weighted avg	0.98	0.98	0.98	820

Figure 11. Classification report

The precision of the model describes that how often the prediction is correct. The recall is also similar to precision, but the condition is that the prediction is accurate by the model. When the average is calculated between these is F1-score, it is mainly used in dealing with imbalances in samples. However, the accuracy is considered closer to evaluating whether the model is balanced or not.

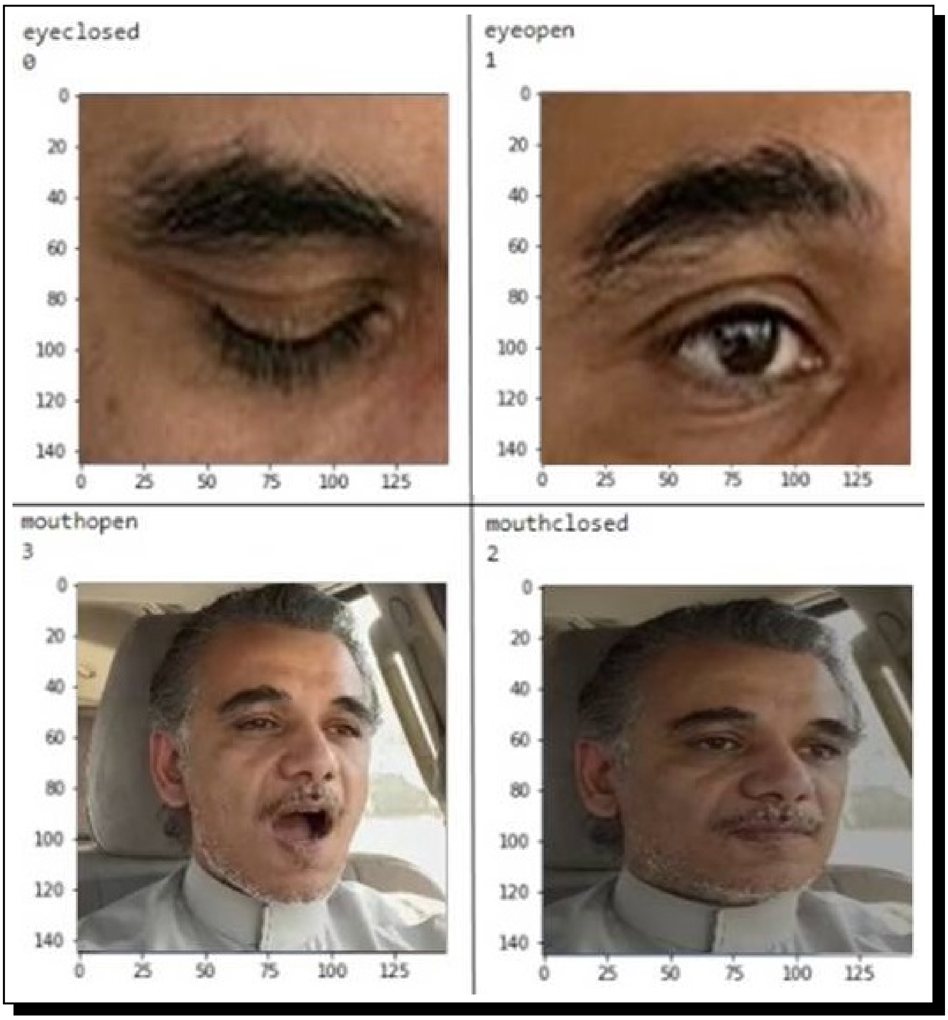

Figure 12. Predictions made by the model

The model makes predictions and gives the output as the label name and class, as seen in Figure 12. The classes go like 0 for eye-opening, 1 for eye closing, 2 for mouth opening, and 3 for mouth closing. The saved model is loaded, and the mixer function from Py-Games is loaded to enable the alarm when the yawning time exceeds, which is kept less than 15 seconds. The video can read the live feed using OpenCV, and the test live-video feed is shown in Figure 13.

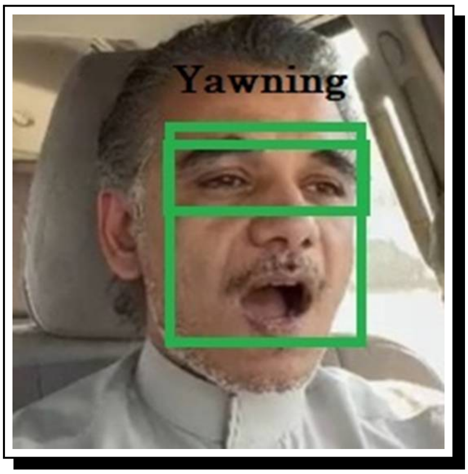

Figure 13. Live-video application of detection of drowsiness

5. Conclusions

The study aims to detect Drowsiness using Dlib and no other pre-trained models like Haar cascade. The CNN model is implemented in this proposed work. A two-way detection implementation is considered using CNN, Dlib, and OpenCV. In this work, face detection is carried out to detect drivers' faces and map their landmarks. The system is trained using data collected with different poses and angles. The work is implemented considering EAR along with thresholding to facilitate the real-time application and future scope of the work. The model is recorded with 98% accuracy, which is considered the highest among the related work [19].

However, the future implementation of work is to consider the CNN model's blinking rate and neck movement detection training to facilitate real-world applications like Uber, Ola, Driverless cars, and many more.

Competing Interests

The author declares that he has no competing interests.

Authors' Contributions

The author wrote, read and approved the final manuscript.

References

[1] T.J. Arnedt, J. Owens, M. Crouch, J. Stahl and M.A. Carskadon, Neurobehavioral performance of residents after heavy night call vs after alcohol ingestion, *Jama* **294**(9) (2005), 1025 – 1033, DOI: 10.1001/jama.294.9.1025.

[2] A. Bhardwaj, W. Di and J. Wei, *Deep Learning Essentials: Your Hands-on Guide to the Fundamentals of Deep Learning and Neural Network Modeling*, Packt Publishing Ltd. (2018).

[3] X. Chai, X. Fu, Z. Gan, Y. Lu and Y. Chen, A color image cryptosystem based on dynamic DNA encryption and chaos, *Signal Processing* **155** (2019), 44 – 62, DOI: 10.1016/j.sigpro.2018.09.029.

[4] V.R.R. Chirra, S. Reddy Uyyala and V.K.K. Kolli, Deep CNN: A machine learning approach for driver drowsiness detection based on eye state, *Revue d'Intelligence Artificielle* **33**(6) (2019), 461 – 466, DOI: 10.18280/ria.330609.

[5] J.P. Cunningham and M.Y. Byron, Dimensionality reduction for large-scale neural recordings, *Nature Neuroscience* **17**(11) (2014), 1500 – 1509, DOI: 10.1038/nn.3776.

[6] E.A. Curran and M.J. Stokes, Learning to control brain activity: A review of the production and control of EEG components for driving brain-computer interface (BCI) systems, *Brain and Recognition* **51**(3) (2003), 326 – 336, DOI: 10.1016/S0278-2626(03)00036-8.

[7] T.A. Dingus, S.K. Jahns, A.D. Horowitz and R. Knipling, Human factors design issues for crash avoidance systems, *Human Factors in Intelligent Transportation Systems* (1998), 55 – 93.

[8] M. Dua, R. Singla, S. Raj and A. Jangra, Deep CNN models-based ensemble approach to driver drowsiness detection, *Neural Computing and Applications* **33**(8) (2021), 3155 – 3168, DOI: 10.1007/s00521-020-05209-7.

[9] H. Ghassemian, A review of remote sensing image fusion methods, *Information Fusion* **32** (2016), 75 – 89, DOI: 10.1016/j.inffus.2016.03.003.

[10] 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* **130** (2018), 400 – 407, DOI: 10.1016/j.procs.2018.04.060.

[11] S.M. Jameel, M.A. Hashmani, M. Rehman and A. Budiman, Adaptive CNN ensemble for complex multispectral image analysis, *Complexity* 2020 (2020), DOI: 10.1155/2020/8361989.

[12] M.W. Johns, A sleep physiologist's view of the drowsy driver, *Transportation Research Part F: Traffic Psychology and Behaviour* **3**(4) (2000), 241 – 249, DOI: 10.1016/S1369-8478(01)00008-0.

[13] S. Kaplan, M.A. Guvensan, A.G. Yavuz and Y. Karalurt, Driver behavior analysis for safe driving: A survey, *IEEE Transactions on Intelligent Transportation Systems* **16**(6) (2015), 3017 – 3032, DOI: 10.1109/TITS.2015.2462084.

[14] A. Kar and P. Corcoran, A review and analysis of eye-gaze estimation systems, algorithms and performance evaluation methods in consumer platforms, *IEEE Access* **5** (2017), 16495 – 16519, DOI: 10.1109/ACCESS.2017.2735633.

[15] O. Khunpisuth, T. Chotchinasri, V. Koschakosai and N. Hnoohom, Driver drowsiness detection using eye-closeness detection, In: *IEEE 12th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS)*, 2016, pp. 661 – 668, DOI: 10.1109/SITIS.2016.110.

[16] E. Lachat, H. Macher, T. Landes and P. Grussenmeyer, Assessment and calibration of a RGB-D camera (Kinect v2 Sensor) towards a potential use for close-range 3D modeling, *Remote Sensing* **7**(10) (2015), 13070 – 13097, DOI: 10.3390/rs71013070.

[17] B. Lantz, *Machine Learning With R: Expert Techniques for Predictive Modeling*, Packt Publishing Ltd. (2019).

[18] D.L. Lau and G.R. Arce, *Modern Digital Halftoning*, Vol. 1, CRC Press, Boca Raton (2018), DOI: 10.1201/9781315219790.

[19] S. Mehta, S. Dadhich, S. Gumber and A.J. Bhatt, Real-time driver drowsiness detection system using eye aspect ratio and eye closure ratio, In: *Proceedings of International Conference on Sustainable Computing in Science, Technology and Management* (SUSCOM), Amity University Rajasthan, Jaipur, India (2019), DOI: 10.2139/ssrn.3356401.

[20] R. Norman, R. Matzopoulos, P. Groenewald and D. Bradshaw, The high burden of injuries in South Africa, *Bulletin of the World Health Organization* **85** (2007), 695 – 702, DOI: 10.2471/BLT.06.037184.

[21] M. Ramzan, H.U. Khan, S.M. Awan, A. Ismail, M. Ilyas and A. Mahmood, A survey on state-of-the-art drowsiness detection techniques, *IEEE Access* **7** (2019), 61904 – 6191, DOI: 10.1109/ACCESS.2019.2914373.

[22] B. Reddy, Y.H. Kim, S. Yun, C. Seo and J. Jang, Real-time driver drowsiness detection for embedded system using model compression of deep neural networks, In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2017, pp. 438 – 445, DOI: 10.1109/CVPRW.2017.59.

[23] V. Sharath, N. Meghana, M. Nayaz, S. Shivakumar and G.L. Sunil, Driver drowsiness detection using haarcascade algorithm, *International Journal of Research in Engineering, Science and Management* **3**(8) (2020), 59 – 60.

[24] C. Shorten and T.M. Khoshgoftaar, A survey on image data augmentation for deep learning, *Journal of Big Data* **6**(1) (2019), Article number: 60, DOI: 10.1186/s40537-019-0197-0.

[25] M. Staubach, Factors correlated with traffic accidents as a basis for evaluating Advanced Driver Assistance Systems, *Accident Analysis & Prevention* **41**(5) (2009), 1025 – 1033, DOI: 10.1016/j.aap.2009.06.014.

[26] C.A. Taylor, J.M. Bell, M.J. Breiding and L. Xu, Traumatic brain injury-related emergency department visits, hospitalizations, and deaths — United States, 2007 and 2013, *MMWR Surveillance Summaries* **66**(9) (2017), 1, DOI: 10.15585/mmwr.ss6609a1.

[27] W.B. Verwey and D.M. Zaidel, Predicting drowsiness accidents from personal attributes, eye blinks and ongoing driving behaviour, *Personality and Individual Differences* **28**(1) (2000), 123 – 142, DOI: 10.1016/S0191-8869(99)00089-6.

[28] C. Ware, *Information Visualization: Perception for Design*, Morgan Kaufmann (2019).

[29] World Health Organization (WHO), *Global Status Report on Road Safety 2015* (2015), URL: http://www.who.int/violence_injury_prevention/road_safety_status/2013/en/index.html.

[30] X. Wu, D. Sahoo and S.C. Hoi, Recent advances in deep learning for object detection, *Neurocomputing* **396** (2020), 39 – 64, DOI: 10.1016/j.neucom.2020.01.085.

[31] R. Yamashita, M. Nishio, R.K.G. Do and K. Togashi, Convolutional neural networks: an overview and application in radiology, *Insights Into Imaging* **9**(4) (2018), 611 – 629, DOI: 10.1007/s13244-018-0639-9.

[32] C.W. You, N.D. Lane, F. Chen, R. Wang, Z. Chen, T.J. Bao, M. Montes-de-Oca, Y. Cheng, M. Lin, L. Torresani and A.T. Campbell, Carsafe app: Alerting drowsy and distracted drivers using dual cameras on smartphones, In: *Proceeding of the 11th Annual International Conference on Mobile Systems, Applications, and Services*, 2013, pp. 13 – 26, DOI: 10.1145/2462456.2465428.

[33] Z.Q. Zhao, P. Zheng, S.T. Xu and X. Wu, Object detection with deep learning: A review, *IEEE Transactions on Neural Networks and Learning Systems* **30**(11) (2019), 3212 – 3232, DOI: 110.1109/TNNLS.2018.2876865.

[34] Y. Zhu and N. Zabaras, Bayesian deep convolutional encoder-decoder networks for surrogate modeling and uncertainty quantification, *Journal of Computational Physics* **366** (2018), 415 – 447, DOI: 10.1016/j.jcp.2018.04.018.