A Real Time Deep Learning Based Driver Monitoring System

Mohamad Faris Fitri Mohd Hanafi[1], Mohammad Sukri Faiz Md. Nasir[2], Sharyar Wani[3], Rawad Abdulkhaleq Abdulmolla Abdulghafor[4], Yonis Gulzar[5], Yasir Hamid[6]

1234Department of Computer Science, Kulliyyah of Information & Communication Technology, International Islamic University Malaysia, Kuala Lumpur, Malaysia

5Department of Management Information Systems, College of Business Administration, King Faisal University, Al-Ahsa, Saudi Arabia

6Information Security and Engineering Technology, Abu Dhabi Polytechnic, Abu Dhabi Polytechnic, Abu Dhabi, UAE

farishanafi@gmail.com, sukrifaiz1995@gmail.com, sharyarwani@iium.edu.my, rawad@iium.edu.my, ygulzar@kfu.edu.sa, yasir.hamid@adploy.ac.ae

Abstract— Road traffic accidents almost kill 1.35 million people around the world. Most of these accidents take place in low and middle-income countries and costs them around 3% of their gross domestic product. Around 20% of the traffic accidents are attributed to distracted drowsy drivers. Many detection systems have been designed to alert the drivers to reduce the huge number of accidents. However, most of them are based on specialized hardware integrated with the vehicle. As such the installation becomes expensive and unaffordable especially in the low- and middle-income sector. In the last decade, smartphones have become essential and affordable. Some researchers have focused on developing mobile engines based on machine learning algorithms for detecting driver drowsiness. However, most of them either suffer from platform dependence or intermittent detection issues. This research aims at developing a real time distracted driver monitoring engine while being operating system agnostic using deep learning. It employs a CNN for detection, feature extraction, image classification and alert generation. The system training will use both openly available and privately gathered data.

Keywords— CNN, Driver Monitoring, Drowsiness, Drowsiness Detector, PERCLOS

I. Introduction

Road accidents kill millions around the world each year. Governments, firms and organizations around the world are scrambling to reduce these statistics. Almost 20% of road accidents are caused by fatigued or distracted drivers. It is projected that distracted drivers could cause up to 65% of accidents by 2035 [1]. A fatigued or distracted driver has considerably increased reaction times compared to an attentive driver. This fact contributes to the increase in the probability of accidents for such drivers. Driver monitoring systems are proven to reduce the accident risks involving fatigued, drowsy or distracted drivers. However, most of the systems proposed are cumbersome to use, require tight integration with the vehicles at factory level, use computers with large power and thermal requirements, require large camera sensors or requires the drivers to wear specialized equipment. Due to these limitations, efforts to make use of smartphones as the processing and sensor hub for these systems have been of interest for the past few years.

Advancements in the field of computer vision and machine learning have enabled the use of artificial neural networks to detect facial cues of a person with a high degree of accuracy and done in real time with a high degree of reliability. This research proposes a real time driver monitoring system using Convolution Neural Networks deployed on a smartphone. The application takes advantage of the inbuilt sensors rather than requiring any external hardware for the detection and alert process.

II. Related Work

Drowsiness detection can be divided into 4 mains aspects: vehicle based, subjective, behavioral, and physiology. Vehicle based input consists of steering wheel status [2], pedal input, car acceleration and speed [3]. The subjective measure uses self-assessment on the sleepiness level, measured majorly using Standford Sleepiness Scale (SSS) and Karolinska Sleepiness Scale (KSS) [4]. Numerous researchers use these scales as a general indicator in assessing drivers' state [5], [7]. Behavioral method measures the action such as eye blink during a set period of time [6], head pose, steering wheel gripping pressure [8], facial detection [9] and eye gaze. Eye gazing method uses eye corners and iris centers to determine the direction of eyes through a linear mapping function while as pupil center detection is based on shape and intensity-based deformable eye using movement decision algorithms [10]. This method is sensitive in accuracy and low in performance when using low-resolution video sequences [11]. Head pose estimation and eye gaze techniques are used to develop POSIT algorithm (Pose from Orthography and Scaling with iTerations) [12]. Physiological measurement is related to brain condition and are considered more accurate. This method uses electrocardiography (ECG) and electroencephalography (EEG) for the measurement process [4]. Some studies use a combination of techniques such as physiological-based and vehicle-based input [13]. The drowsiness of a driver can be accurately measured using several ways, most prominent of all is the PERCLOS measurement [14].

Deep learning based on convolutional neural networks have been heavily used for computer vision. A CNN based system has been proposed in [15], using a camera positioned on the dashboard of the vehicle as input and Nvidia Jetson TK1 board as the processing node. However, this technique is computationally intensive and thus difficult to run in real time on a small power budget. To overcome this problem, the researchers propose a compression algorithm of deep learning model has been proposed which distills the neural network. Distillation of a neural network is transferring knowledge from a large model to a small model. Small models are not suitable to be trained directly as they do not converge easily. A fusion of facial information using Deep Belief Networks (DBN) has also been proposed as a technique for driver monitoring. This is to overcome the lack of generalization capability of contemporary methods such as eye or mouth detection alone, observing an accuracy of 96.7% [16]. The use of Haar Cascade detection together with template matching using OpenCV to detect the eyes of the driver have been proposed for drowsiness detection [17], [18]. In essence, this technique detects and uses the blink behavior as an identifier for drowsiness.

Hybrid approaches have also been proposed for higher detection accuracy. A combination of smartphone and wearable device to detect eye blinking and heart rate has been presented in [17]. The combined information is compared to the ranges predetermined for a drowsy driver. Reference [19] uses the electrodes in their experiment with a few variations such as reducing the EEG electrode to enhance the wearability. Another system combines input from an IR detector, an accelerometer, a thermistor, IR LED and a phototransistor into a microcontroller [20].

Detection and measurement of PERCLOS for drivers wearing spectacles is a major challenge for detection systems. This is due to the glare present on the lenses of the glasses during daylight which is a result of outside reflections and ambient light. Another hurdle is the effectiveness of small front facing cameras found on smartphones to gather light in low light conditions [21]. Infrared cameras have been proposed as potential solutions to these problems [22]. However, such cameras are not available in most smartphones and there is also a lack of API's for developers to take advantage of them, where available

Processing power is yet another major problem in driver drowsiness detection systems. Artificial neural networks typically used for this purpose are generally resource intensive and require powerful processing systems to work reliably in real time. Some researchers used powerful personal computers [9], while others used embedded development boards specialized for deep learning to cope with this limitation [15]. Many of the latest smartphone include specialized ASICs called Neural Processing Units (NPU) that can perform these calculations much faster at a smaller power budget. Manufacturers have also made APIs and SDKs available for developers to make use of this part on the SoC [21].

Besides image detection and classification using the drivers physical state, sensor data can be integrated into the vision-based models. Researchers in compare vision-based modelling using CNN and compare it with vision and sensor-based driver monitoring using LSTM-RNN. The mixed data model significantly increases the system performance to 85% accuracy compared to 62% of image only modeling [23].

Distraction system can use feature rich data by collecting data from multiple types of sensors such as physiological sensors and visual sensors. Statistical tests in [24] reveal that the most corelated feature for detecting driver distraction point to emotional activation and facial action. The researchers used seven classical machine learning (ML) and deep learning (DL) models, confirming XGB with the highest F1-score of 79% using 60-second windows of AUs as input. The score recorded an enhancement up to 94% during complete driving sessions. Spectro-temporal ResNet, scored the highest F1-score of 0.75 when classifying segments and 0.87 when classifying complete driving sessions among the used deep learning models.

III. Experimental Setup

A. Dataset

The dataset used is the Closed Eyes in the Wild (CEW) dataset [25]. The dataset contains 3876 images of left and right eyes in various open or closed states, refer to Table I. This was extracted from faces in various lighting conditions. The images are then separated into left/right and open/close labels. Then, they undergo normalization by pre-processing into greyscale images. This is done by dividing with their mean RGB values and then downscaled into 24x24 pixels resolution. This dataset is then separated into training and testing sets with a ratio of 3:1.

Fig. 1 System Working Flow

TABLE I
SAMPLE LABELLED IMAGES FROM THE DATASET

Open Right Eye	Open Left Eye	Closed Right Eye	Closed Left Eye

B. *Proposed CNN Architecture*

Our model (Fig 2) is built upon the foundation of a normal Convolutional Neural Network. It consists of 4 convolution layers, and two dense layers. In the first layer, we have 32 3x3 filters and we used the Rectified Linear Unit (ReLU as the activation function). In the second layer we have 24 3x3 filters together with 2x2 max-pooling filters and applied a dropout of 0.25. This was done to reduce the likelihood that we are overfitting our data to our model. Overfitting is when the model memorizes the data instead of learning it. We also kept this in mind when deciding the number of epochs to run our model later on. The next layer consists of 64 3x3 filters and used the same ReLU activation function. The layer after that consists of again 64 3x3 filters with 2x2 max-pooling filters and a dropout of 0.25 again.

We then flatten the output of the previous layers before passing to the 512-node dense layer. In this layer the activation function is still ReLU while the dropout value is increased to 0.5. Lastly this network is activated with a sigmoid function before throwing out the output. The epoch is set at 25 to balance between training our data while not overfitting it to the model.

With the model set to use the SGD optimizer algorithm, we ran the model on our test set to validate it and achieved an accuracy of 90%.

In an effort to improve the accuracy of our model, we trained the model multiple times with different optimizer algorithms and observed the findings from there. We kept everything else a constant and only changed the optimizer algorithms. We tested the validation loss, validation accuracy and training accuracy of the model. While using the SGD algorithm, we noticed that the validation loss is still quite high even after 25 epochs. The validation accuracy and training accuracy also have not plateaued after 25 epochs.

C. *Mobile App*

The mobile app is built using the Flutter framework which is a multiplatform framework that works for Android and IOS. The app works by taking an 848x480 video stream from the camera and then tracks the user's face. The app then performs real time landmark extraction. From there, the app tracks the user's eyes and performs real-time inferencing based on the pretrained model preloaded into the app's assets. By leveraging the NNAPI and ML Kit framework available in Android 8.1 and up, this was able to be done in real time with satisfactory responsiveness as it uses the DSP on the device.

The PERCLOS is measured and if a threshold of less than 12 blinks per minute is exceeded the app alerts the user. Since the app uses the front camera of the device, the field of view is limited to the lens of the camera. Front cameras on smartphones are typically RGB cameras that capture colour using a Bayer filter. However, this also results in less light taken in and impacts the performance in low light scenarios.

D. *Optimization Algorithms*

To gauge the most suitable optimizer for our purposes, we compared four optimizer algorithms. They are SGD, Adam, Adadelta, and Adagrad. All other variables are constant with only the optimizer being changed. The training-validation loss and accuracy for the optimizers have been presented in presented in Fig 3 – Fig 10.

IV. RESULTS

The plots reveal that the models do not underfit or overfit the data with the exception of the SGD optimizer. It shows relatively high loss and low accuracy while not plateauing even after 25 epochs. The SGD optimized model achieved a 90% test accuracy, while the Adagrad, Adam and Adadelta optimized model achieved 96% test accuracy. Based on these graphs, we decided to use the Adam optimized model to be used as our trained model. This model is then converted to Tensorflow Lite format for use on mobile devices.

Fig. 2 Proposed CNN Architecture

TABLE III
ARCHITECTURE COMPARISON

Architecture	Accuracy	Computational Requirement
DBN	96.7%	High
CNN (Adam)	96%	Moderate

Our model managed to achieve an almost similar accuracy as the system developed by [21], refer to Table II. The system in question however uses DBN or Deep Belief Network which is more computationally intensive compared to our CNN model. Our model was also able to achieve this accuracy while running on a Snapdragon 845 smartphone in real time instead of using more powerful computers or dedicated sensors.

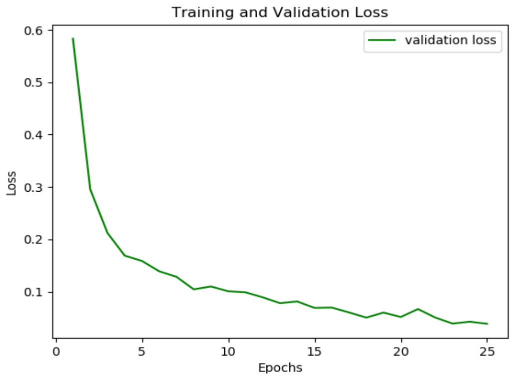

Fig. 3 Adam Optimizer Loss

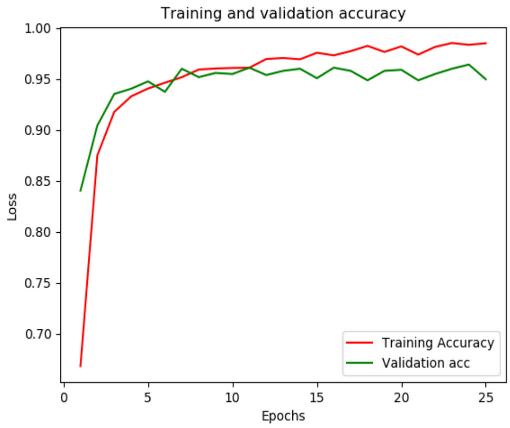

Fig. 4 Adam Optimizer Accuracy

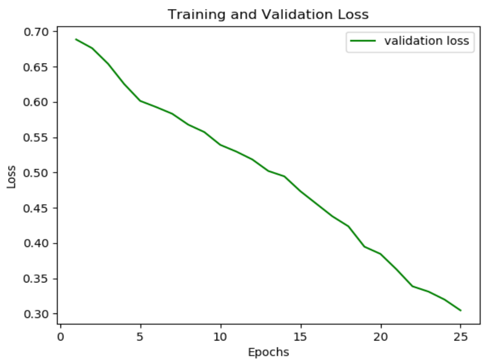

Fig. 5 SGD Optimizer Loss

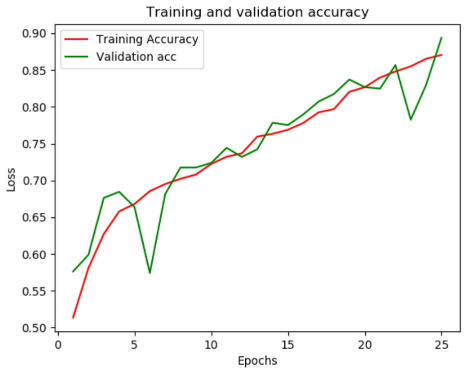

Fig. 6 SGD Optimizer Accuracy

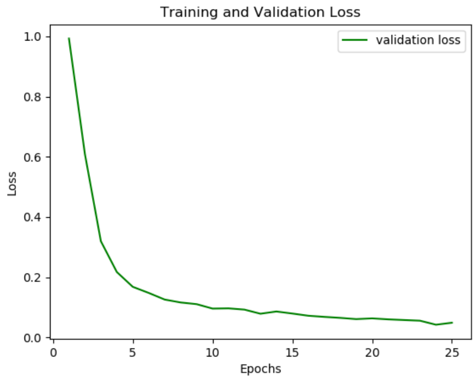

Fig. 7 Adagrad Optimizer Loss

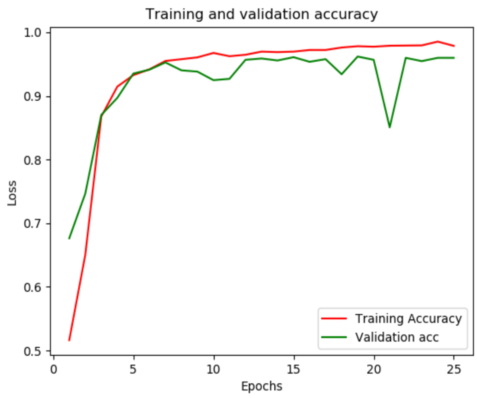

Fig. 8 Adagrad Optimizer Accurcay

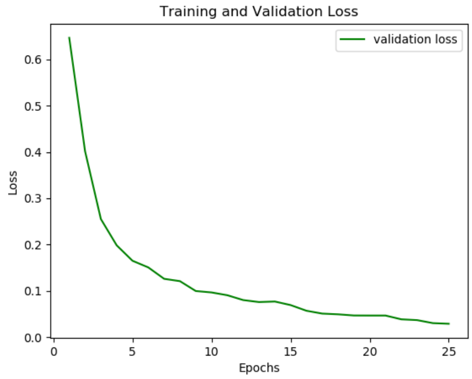

Fig. 9 Adelta Optimizer Loss

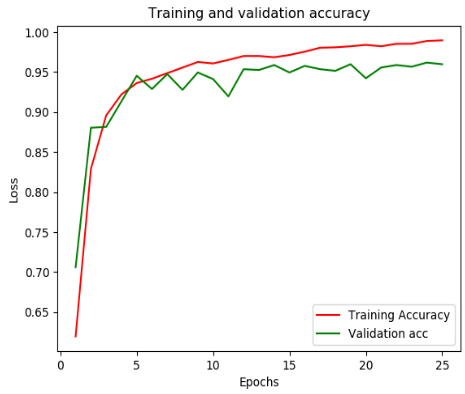

Fig. 10 Adelta Optimizer Accuracy

In practical testing of our smartphone application that is loaded with the trained model, the application manages to detect and classify the user as sleepy or not sleepy based on the parameter described before which is 12 blinks per minute. The polling window for inferencing and deciding whether the user is sleepy is variable, and we set ours at 3 blinks per 15 seconds.

V. Conclusion & Future Work

This paper presented a deep learning based real time distracted driver monitoring system deployable on a mobile app. CNN achieved an accuracy of 96%, similar to a Deep Belief Network system, while being computationally less expensive. PERCLOS was used as the quantification measure for distraction or drowsiness detection.

The system performance can be potentially improved by using models such as Mobile Net and using IR cameras available on the front of smartphones for biometric face unlocks. This camera can be utilized instead of the main front facing camera and could yield much better performance in challenging light conditions.

PERCLOS can be combined with detection of other indicators such as face pose, gaze detection and yawn detection. A fusion of these metrics could result in a more accurate detection and reduce false positives.

REFERENCES

[1] R. Manoharan and S. Chandrakala, "Android OpenCV based effective driver fatigue and distraction monitoring system," in *Proceedings of the International Conference on Computing and Communications Technologies, ICCCT 2015*, Oct. 2015, pp. 262–266, doi: 10.1109/ICCCT2.2015.7292757.

[2] M. Chai, shi wu Li, wen cai Sun, meng zhu Guo, and meng yuan Huang, "Drowsiness monitoring based on steering wheel status," *Transp. Res. Part D Transp. Environ.*, vol. 66, pp. 95–103, Jan. 2019, doi: 10.1016/j.trd.2018.07.007.

[3] A. D. McDonald, J. D. Lee, C. Schwarz, and T. L. Brown, "A contextual and temporal algorithm for driver drowsiness detection," *Accid. Anal. Prev.*, vol. 113, pp. 25–37, Apr. 2018, doi: 10.1016/j.aap.2018.01.005.

[4] M. Awais, N. Badruddin, and M. Drieberg, "A hybrid approach to detect driver drowsiness utilizing physiological signals to improve system performance and Wearability," *Sensors (Switzerland)*, vol. 17, no. 9, Sep. 2017, doi: 10.3390/s17091991.

[5] S. Noori and M. Mikaeili, "Driving drowsiness detection using fusion of electroencephalography, electrooculography, and driving quality signals," *J. Med. Signals Sens.*, vol. 6, no. 1, pp. 39–46, Jan. 2016, doi: 10.4103/2228-7477.175868.

[6] A. M. Rumagit, I. A. Akbar, and T. Igasaki, "Gazing time analysis for drowsiness assessment using eye gaze tracker," *Telkomnika (Telecommunication Comput. Electron. Control.*, vol. 15, no. 2, pp. 919–925, Jun. 2017, doi: 10.12928/TELKOMNIKA.v15i2.6145.

[7] X. Wang and C. Xu, "Driver drowsiness detection based on non-intrusive metrics considering individual specifics," *Accid. Anal. Prev.*, vol. 95, pp. 350–357, Oct. 2016, doi: 10.1016/j.aap.2015.09.002.

[8] Y. Chellappa, N. N. Joshi, and V. Bharadwaj, "Driver fatigue detection system," in *2016 IEEE International Conference on Signal and Image Processing, ICSIP 2016*, Mar. 2017, pp. 655–660, doi: 10.1109/SIPROCESS.2016.7888344.

[9] C. Y. Lin, P. Chang, A. Wang, and C. P. Fan, "Machine Learning and Gradient Statistics Based Real-Time Driver Drowsiness Detection," Aug. 2018, doi: 10.1109/ICCE-China.2018.8448747.

[10] A. G. Mavely, J. E. Judith, P. A. Sahal, and S. A. Kuruvilla, "Eye gaze tracking based driver monitoring system," in *IEEE International Conference on Circuits and Systems, ICCS 2017*, Mar. 2018, vol. 2018-January, pp. 364–367, doi: 10.1109/ICCS1.2017.8326022.

[11] I. F. Ince and J. W. Kim, "A 2D eye gaze estimation system with low-resolution webcam images," *EURASIP J. Adv. Signal Process.*, vol. 2011, no. 1, pp. 1–11, Aug. 2011, doi: 10.1186/1687-6180-2011-40.

[12] I. H. Choi and Y. G. Kim, "Head pose and gaze direction tracking for detecting a drowsy driver," in *2014 International Conference on Big Data and Smart Computing, BIGCOMP 2014*, 2014, pp. 241–244, doi: 10.1109/BIGCOMP.2014.6741444.

[13] J. Vicente, P. Laguna, A. Bartra, and R. Bailón, "Drowsiness detection using heart rate variability," *Med. Biol. Eng. Comput.*, vol. 54, no. 6, pp. 927–937, Jun. 2016, doi: 10.1007/s11517-015-1448-7.

[14] H. Singh, J. S. Bhatia, and J. Kaur, "Eye tracking based driver fatigue monitoring and warning system," 2011, doi: 10.1109/IICPE.2011.5728062.

[15] 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 *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, Aug. 2017, vol. 2017-July, pp. 438–445, doi: 10.1109/CVPRW.2017.59.

[16] L. Zhao, Z. Wang, X. Wang, and Q. Liu, "Driver drowsiness detection using facial dynamic fusion information and a DBN," *IET Intell. Transp. Syst.*, vol. 12, no. 2, pp. 127–133, Mar. 2018, doi: 10.1049/iet-its.2017.0183.

[17] S. S. Kulkarni, A. D. Harale, and A. V. Thakur, "Image processing for driver's safety and vehicle control using raspberry Pi and webcam," in *IEEE International Conference on Power, Control, Signals and Instrumentation Engineering, ICPCSI 2017*, Jun. 2018, pp. 1288–1291, doi: 10.1109/ICPCSI.2017.8391917.

[18] K. U. G. S. Darshana, M. D. Y. Fernando, S. S. Jayawadena, and S. K. K. Wickramanayake, "Riyadisi - Intelligent driver monitoring system," in *International Conference on Advances in ICT for Emerging Regions, ICTer 2013 - Conference Proceedings*, 2013, p. 286, doi: 10.1109/ICTer.2013.6761200.

[19] I. Belakhdar, W. Kaaniche, R. Djemal, and B. Ouni, "Single-channel-based automatic drowsiness detection architecture with a reduced number of EEG features," *Microprocess. Microsyst.*, vol. 58, pp. 13–23, Apr. 2018, doi: 10.1016/j.micpro.2018.02.004.

[20] A. Bhaskar, "EyeAwake: A cost effective drowsy driver alert and vehicle correction system," in *Proceedings of 2017 International Conference on Innovations in Information, Embedded and Communication Systems, ICIIECS 2017*, Jan. 2018, vol. 2018-January, pp. 1–6, doi: 10.1109/ICIIECS.2017.8276114.

[21] L. Xu, S. Li, K. Bian, T. Zhao, and W. Yan, "Sober-drive: A smartphone-assisted drowsy driving detection system," in *2014 International Conference on Computing, Networking and Communications, ICNC 2014*, 2014, pp. 398–402, doi: 10.1109/ICCNC.2014.6785367.

[22] B. Alshaqaqi, A. S. Baquhaizel, M. E. A. Ouis, M. Boumehed, A. Ouamri, and M. Keche, "Vision based system for driver drowsiness detection," in *Proceedings of the 2013 11th International Symposium on Programming and Systems, ISPS 2013*, 2013, pp. 103–108, doi: 10.1109/ISPS.2013.6581501.

[23] F. Omerustaoglu, C. O. Sakar, and G. Kar, "Distracted driver detection by combining in-vehicle and image data using deep learning," *Appl. Soft Comput.*, vol. 96, p. 106657, Nov. 2020, doi: 10.1016/J.ASOC.2020.106657.

[24] M. Gjoreski, M. Z. Gams, M. Luštrek, P. Genc, J. U. Garbas, and T. Hassan, "Machine Learning and End-to-End Deep Learning for Monitoring Driver Distractions from Physiological and Visual Signals," *IEEE Access*, vol. 8, pp. 70590–70603, 2020, doi: 10.1109/ACCESS.2020.2986810.

[25] F. Song, X. Tan, X. Liu, and S. Chen, "Eyes closeness detection from still images with multi-scale histograms of principal oriented gradients," *Pattern Recognition.*, vol. 47, no. 9, pp. 2825–2838, Sep. 2014, doi: 10.1016/j.patcog.2014.03.024.