Real-time Driver Drowsiness Detection using Deep Learning

Md. Tanvir Ahammed Dipu[1], Syeda Sumbul Hossain[2], Yeasir Arafat[3], Fatama Binta Rafiq[4]

Department of Software Engineering, Daffodil International University

Dhaka, Bangladesh

Abstract—**Every year thousands of lives pass away worldwide due to vehicle accidents, and the main reason behind this is the drowsiness in drivers. A drowsiness detection system will help to reduce this accident and save many lives around the world. To defend this problem, we propose a methodology based on Convolutional Neural Networks (CNN) that illustrates drowsiness detection as a task to detect an object. It will detect and localize whether the eyes are open or close based on the real-time video stream of drivers. The MobileNet CNN Architecture with Single Shot Multibox Detector is the technology used for this object detection task. A separate algorithm is used based on the output given by the SSD_MobileNet_v1 architecture. A dataset that consists of around 4500 images was labeled with the object's face yawn, no-yawn, open eye, and closed eye to train the SSD_MobileNet_v1 Network. Around 600 randomly selected images are used to test the trained model using the PASCAL VOC metric. The proposed approach is to ensure better accuracy and computational efficiency. It is also affordable as it can process incoming video streams in real-time and does not need any expensive hardware support. There only needs a standalone camera to be implemented using cheap devices in cars using Raspberry Pi 3 or other IP cameras.**

Keywords—Deep learning; drowsiness detection; object detection; MobileNets; Single Shot Multibox Detector

I. INTRODUCTION

Drowsiness while driving tends to vehicle crashes and accidents. Many people die in car collisions every year due to fatigued driving that results from sleeping deprivation, intoxication, drug and alcohol abuse, exposure to heat or alcohol. Automobile manufacturers [8] such as Tesla, Mercedes-Benz, and others have various features for driving assistance such as warning of lane deviation, emergency braking systems, variable cruise control, and aid steering. These innovations have assisted drivers in avoiding the incidence of collisions. Samsung has investigated the attention level of a driver by reading facial characteristics and patterns [10]. At the same time, most of these technologies are proprietary and restricted to high-end cars. These drowsiness identification processes can be divided depending on some of the methods like based on the context of vehicle, behavioral & physiological [39]. Multiple methods for the identification of drowsiness have been developed in the past. Based on vehicle-based, drowsiness identification processes are performed for monitoring the lane switches, steering wheel spin, velocity, compressions on the accelerator pedal. These approaches include measurement of driver physiological signals [11], performance assessment based on vehicles [12], and recording

behavior [13]. The bio-signal measurement method demonstrated the highest ability to detect driver drowsiness among these techniques: unlike the other two approaches, it relies solely on the state of the driver. Based on behavior like specific eye closure, yawn, and head posture, drowsiness identification processes need a camera. Another step based on physiological drowsiness identification processes works by monitoring the tiredness to relate between their physiological signals like EOG (electrooculogram) and ECG (Electrocardiogram) [14,15]. The limitation of drowsiness identification processes using the physiological method is that the diver needs to contain electrodes on their body [14]. There is a substantial restriction based on vehicle-based drowsiness identification [15], such as they are prone to forces connected to drivers and vehicles, road situations. There are many techniques stated in different literature those has some restrictions and several benefits.

The aim of this research is to propose a cost-effective procedure to identify the drowsiness among divers while driving vehicles. To develop the drowsiness detector application, we have used a CNN architecture. This work can be divided into two parts based on the main contribution: (a) Convolutional neural networks to identify right drowsiness identification processes based on object detection, (b) and drowsy datasets to help the researchers for drowsiness identification method.

The rest of the paper is organized in the following style: Literature Review is portrayed in Section II that followed by Research Methodology and Results & Discussion in Section III and Section IV, respectively. The final section depicts our Contribution and Limitations.

II. LITERATUE REVIEW

Eyelid closing has been a much more reliable predictor of drowsiness. Many of the systems has built ought to rely on eyelid closing for driver drowsiness detection even though the other behavior is also a predictor like faster blinking time, sneezing, a slow movement of the eyelid, repeated blinking, set eyes, and sagging pose. Many works of literature have been proposed in this field to predict drowsiness using normal cameras, Infrared (IR) cameras, and stereo cameras. A literature review [27] has been performed to summarize the different drowsiness detection systems and tools. Dwivedi et al. [19] was developed a model using CNN to find drowsiness. In this approach, CNN-based representation feature learning was used and achieved 78% accuracy. To reduce the traffic injuries related to driver drowsiness, the Specialized Driver

Assistance System was proposed by Alshaqaqi et al. [20]. An algorithm was suggested to locate, map, and evaluate face and eyes to test PERCLOS to diagnose driver drowsiness. Said et al. [23] suggested an Eye-tracking-based driver drowsiness system. The method detects the drowsiness of the driver in this task and rings the alarm to warn drivers. Viola Jones's model was used to detect the area of the face and eye in this work. In indoor tests, it provided 82% accuracy and 72.8% accuracy for the outdoor setting.

Mehta et al. [26] have created a mobile app that can detect facial landmarks and then compute the Eye Aspect Ratio (EAR) and Eye Closure Ratio (ECR) to predict driver drowsiness with an accuracy of 84% based on machine learning models. Smart glass has been created by a start-up Ellcie-Healthy [30] that incorporates a somnolence monitoring technology by providing blink detection, Eye recording, and control of vital signs. By beeping and thereby telling the driver to take a break, the smart glass tracks these inputs and offers somnolence interference. Combination strategies involve multiple sensors on a single device, such as infrared, cameras, and heart rate monitors, to produce outstanding performance. These tools are very costly and require the setting up of proprietary solutions.

Mandal et al. [24] have been proposed a vision-based fatigue identification method for bus driver monitoring. In this work, AHOG and SVM are used respectively for head-shoulder detection and driver detection. They used the OpenCV face detector for face detection and the OpenCV eye detector for eye detection. To learn the eye structure, Spectral Regression Embedding was used, and a new approach was introduced for calculating eye openness. Fusion was used to fuse the features created by two I2R-ED and CV-ED eye detectors. For drowsiness identification, PERCLOS was determined. To detect yawning on the YawDD and NTHU-DDD databases, Xie et al. [25] used transfer learning and sequential learning from yawning video clips. This method was more accurate and stable in terms of adjustments in the direction and orientation of the face to the camera. A vision-based MultiTask Driver Control System was suggested by Celona et al. [21] to examine the eyes, mouth, and location of the head simultaneously to predict the degree of drowsiness.

Jabbar et al. [10, 28] introduced a concept for detecting driver drowsiness for android apps focused on deep learning. They developed a model here that is focused on the identification of facial landmark points. The first images are derived from video frames here, and then the Dlib library was used to extract coordinate points for landmarks. These landmark coordinate points are given as input to the multi-layer perceptron classifier. The classifier classifies these points depending on either drowsy or non-drowsy. Shakeel et al. [22] used the MobileNet-SSD architecture to train 350 images for a custom dataset. The model was capable of achieving a 0.84 Mean Average Precision. As the algorithm could be implemented on an Android device, and the camera stream could be classified in real-time, the method was cost-effective and successful. A Long-term Multi-granularity Deep structure was suggested by Jie Lyu et al. [29] to diagnose driver drowsiness with an accuracy of 90.05 %. However, due to its difficulty, the system was not able to be implemented on mobile devices.

III. METHODOLOGY

The overall research methodology is undelineated in this section.

A. Algorithm

It perceives that drowsiness detection is an object detection task. We use images from incoming video streams. From several techniques of deep learning, we use MobileNets [32], a lightweight convolutional neural network architecture, along with the Single Shot Multibox Detector (SSD) [33] system that is on top of the MobileNet architecture to experiment.

- Convolutional Neural Network (CNN)

The suggested system for detecting driver drowsiness is used in the Convolutional Neural Network (CNN) [40]. The pre-processing for CNN is much meagerer compared to others classification algorithms. CNN is a mathematical technique that usually consists of three kinds of pooling, convolution, and fully connected layers. The first two pooling layers are convolution handle extraction of features, and the third one is a completely connected layer, maps the characteristics extracted, such as classification, into the final output. In CNN, which consists of a set of logical operations, such as convolution, A technical method of linear motion, the convolution layer plays a key role. In image data, two-dimensional arrays of pixel values are processed, and at each image location, a small parameter grid called the kernel and function extractor for optimization is added, making CNN's highly effective for image processing since a feature can appear anywhere in the image. Extracted functionality will get more complex hierarchically and gradually as one layer feeds the output into the next layer. The way parameters such, as kernels are optimized is called planning. This can be achieved by a back-propagation and gradient descent optimization algorithm to minimize the discrepancy between outputs and ground truth marks. Fig. 1 represents the CNN architecture.
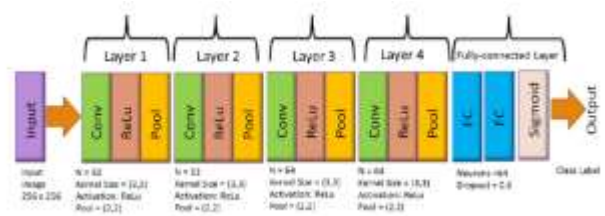

Fig. 1. CNN Architecture.

- Single Shot Multibox Detector (SSD)

For our task, we use MobileNets [32], a lightweight convolutional neural network architecture, along with the Single Shot Multibox Detector (SSD) [33] system that is on top of the MobileNet architecture. There are two types of algorithms used to detect typical objects. To define regions where objects can be found, RCNN [42] and Faster RCNN [43] algorithms use a two-step approach to identify objects only in certain regions. Algorithms such as SSD [33] and YOLO [41] use a completely convolutionary method, on the

other hand, through which the network will locate all items in a picture via the ConvNet in one pass. Usually, the algorithms for the region proposal have slightly better precision but are slower to run, whereas single-shot algorithms are more powerful and have satisfying precision. Fig. 2 represents the SSD architecture.
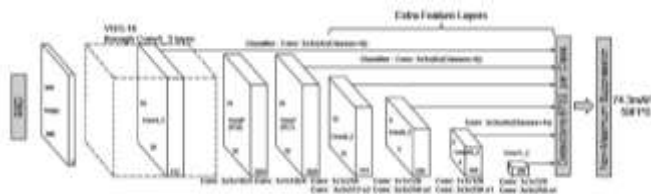

Fig. 2.   SSD Architecture.

The SSD has two elements, a backbone model and an SSD head. The Backbone model is usually a pre-trained image classification network as a feature extractor. This network is educated on ImageNet such as ResNet which excludes the entirely connected classification layer. Therefore, we have a deep neural network that can infer semantic meaning from the input image while preserving the spatial structure in the image at a lower resolution. We will clarify the later feature and feature map. The SSD head is only one or more convolutionary layers attached to this backbone, and the outputs are represented as the bounding boxes and object groups in the final layer activation spatial role. There are some important parameters in SSD. The SSD uses a grid (see Fig. 3(a)) to divide the image rather than a sliding window, and each cell of the grid needs to detect the objects in that picture region. Object identification involves simply predicting the class and location of the object within the sector. We admire it as the context class if no object is present, and the location is ignored. Each grid cell is output to the position and the form of the entity it carries.

Fig. 3.   (a) Grid View (b) Anchor Boxes (c) Higher and Lower Anchor Boxes.

If many objects of different shapes exist in one grid cell, the anchor box (see Fig. 3(b)) and the responsive area are considered. There may be separate anchor boxes in the SSD for each grid cell. These anchor boxes are predefined, and each of them is responsible for the size and shape of a grid cell. The face in Fig. 3(c), for example, has the big anchor box while the eyes correspond to the small box. In training, the SSD framework utilizes a mapping step to align the correct anchor box with the bounding box inside an image of each ground truth entity. The anchor boxes with a greater degree of overlap with an object are ultimately responsible for estimating the class of that object and its position. This property is used for preparing the network for detecting objects and their locations. In reality, an aspect ratio and a zoom level describe any anchor box. The objects are not always square those could be

narrower, broader, or in varying degrees. With predefined aspect ratios, the SSD architecture requires the anchor boxes to account for this. The parameter ratios are used to determine the various aspect ratios of the anchor boxes at each scale level associated with each grid cell. It is not needed to have the same size as the grid cell for the anchor boxes. Inside the cell, we may select smaller or larger objects. The zoom parameter is used to determine how much for each grid cell the anchor boxes need to be scaled up or down.

The receptive field is defined as the area that a specific CNN function looks at in the input space. We have used "feature" and "activation" here and treat them at the corresponding position as the linear combination of the previous layer. Features on different layers reflect different area sizes in the input picture due to the convolution operation. The scale defined by a function grows bigger while going deeper. We begin with the bottom layer (5x5) in this example below and then add a convolution resulting in the middle layer (3x3), where one attribute represents a 3x3 area of the input layer. And then, add the middle layer convolution and get the upper layer (2x2) where each attribute on the input image corresponds to a 7x7 area. This type of green and orange 2D array is often referred to as feature maps that correspond to a group of features generated in a sliding window fashion by applying the same feature extractor at different input map locations. The features in the same map of characteristics have the same receptive area and aim at different positions with the same pattern.

The core concept of the SSD architecture helps to detect objects at varying scales and output a closer bounding box. As in the ResNet34 backbone outputs an input image function map of 256 7x7. The easiest solution is to define a 4x4 grid to add a convolution to this function map and transform it to 4x4. In reality, this method will function to some degree as the concept of YOLO. The additional step taken by SSD is to attach more convolutional layers to the map of the backbone feature and generate an object detection result for each of these convolution layers. Because earlier layers with a smaller receptive field will reflect smaller objects, earlier layer predictions help with the smaller object. Fig. 4 represents the Convolutional Neural Network feature maps
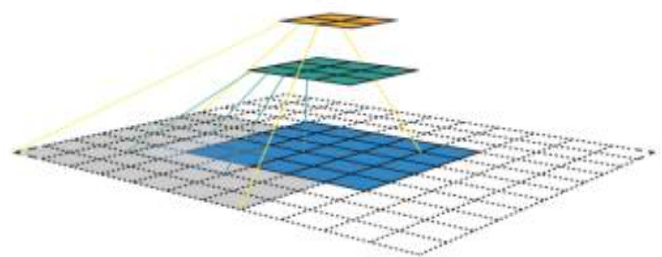

Fig. 4.   Convolutional Neural Network Feature Maps.

- MobileNet

The architecture of MobileNets is designed for mobile-based applications, is used for depth-wise separable convolutions, and builds lightweight deep neural networks. Except for the first layer, the MobileNet is a separable

convolution [31]. A complete convolutional layer is the first layer. Batch normalization and ReLU non-linearity are accompanied by all the layers. The final layer, however, is a completely related layer without any non-linearity and feeds for classification to the softmax. For downsampling, stridden convolution is used for both deep convolution and the first fully convolutionary layer. The total number of layers for MobileNet is 28, considering convolution as different layers in-depth and stage.

The MobileNet-SSD_v1 [34] framework is capable of detecting multiple objects from an image. We trained it to see a human face for our drowsiness detection task to notice that the diver is yawn or no yawn and the eye open or close. Yawn, no yawn, eye open and close were considering four separate classes. For this experiment, a regular camera is used for the incoming video stream. The technique suggested is exactly presented in Fig. 5.

We observed that the longest length of a blink found equals 7.5 inches from the camera in the training model. From the video stream, we get an image frame, and this function is called image frames processed per second (FPS). The model would declare the driver to be drowsy if any of the driver's eyes are found close for ten consecutive frames in the incoming video stream, and an alarm will raise to wake the driver. The above approach uses only a single convolutional neural network and, thus, the total complexity of the system is much smaller. Now we need to see how well this recommended approach performs under driving environments in real life.
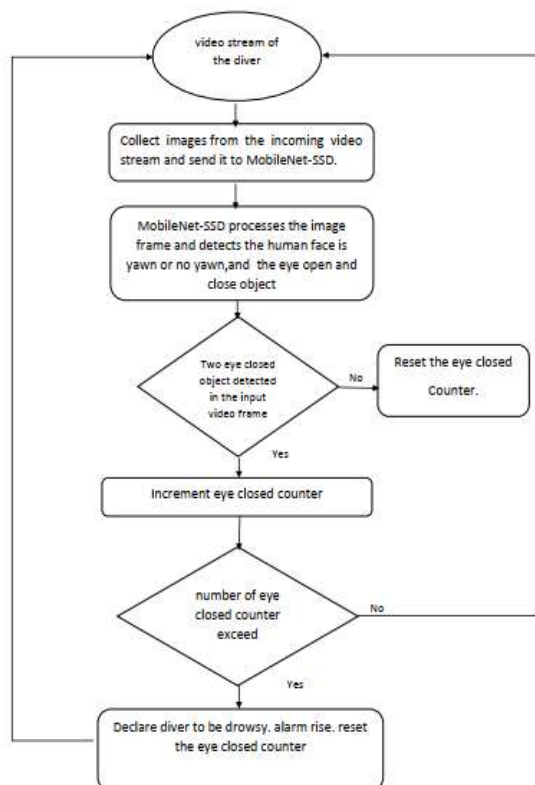

Fig. 5.   Drowsiness Detection System Methodology.

B. Data Collection

We have collected and annotated a customized data set to train our model in the MobileNet-SSD system. The annotated image data includes human faces with opened and closed eyes. Our collected data sets consist of images from a few publicly accessible online databases released by reputed users. Fig. 5 represents some dateset.

IV.   OPEN AND CLOSE EYES

The dataset open and closed eyes were collected from the open-source database. This dataset contains all the human eyes that were closed and open. All pictures are available in JPG format. There are 1234 images, which is nearly 10 MB, without annotations. We have annotated all the images in this dataset.

V.   YAWNING AND NO YAWN WHILE DRIVING

The data source of yawning and no-yawning during driving is given by the University of Ottawa, Canada [35]. The data set includes two types of videos that are captured from two separate camera positions inside the vehicle. For the first set of data, the device is installed on the dashboard of the car, and for the second set of data, the device is installed just underneath the front mirror of the car. There are both male and female drivers in these videos. In these videos, the driver is wearing glasses, and some of the drivers are not wearing glasses. There is also an available laughing driver, a yawning driver, a not-yawning driver, and some drivers are looking around. We collected about 1234 frames from all the videos for this dataset. After extracting the frame from the videos, we annotated all the images with labels eyes open, close, yawn, and no-yawn.

VI.   CUSTOM USED IMAGES

The photos used are freely available and allowed for re-use. The main objective of using these photos is to improve our classification model. This dataset is versatile with different poses and lighting environments. All Pictures are available in JPG format. There are 1900 images, which is nearly 25 MB, are available in this dataset. Although these photos are collected from several online sources, no annotations are available for this dataset. We collected about 250 frames from all the videos for this dataset. After extracting the frame from the videos, we annotated all the images with labels eyes open, close, yawn, and no-yawn.
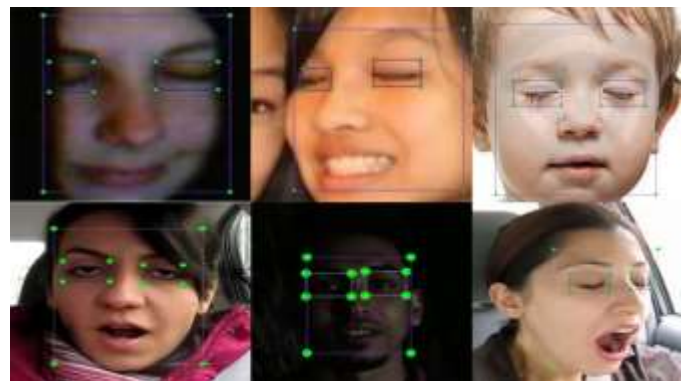

Fig. 6.   Dataset Images.

From the image data Fig. 6, one can see that the Drowsy dataset was already collected to combine images from a wide range of positions, perspectives, and lighting conditions to create a balanced dataset to ensure high precision and generalization of the object detection system. Although this dataset contains very few numbers of low-light images, as of now, the objective has been limited to identification drowsiness in daytime conditions.

A. Dataset Training

We have used Tensorflow API to train our model. As TFOD API can read only TFRecord file format, we need to convert our dataset into this format. First, we need an RGB image dataset in jpeg or png format; and create the bounding boxes for the image and mention the classes for those boxes. With LabelImg, we hand-labeled them manually. LabelImg is a Python-written graphical image annotation application with a good user interface. It also supports Python 2 and Python 3. After completing annotation, we convert our dataset XML to CSV to create the TFrecords. We use around 4500 files for training (train.records) and 700 files for testing (test.records). Although the generated dataset did not contain sufficient training images to train the object detection framework, the transfer learning principle has been used [36, 37]. For training our model, we take the TensorFlow Object Detection Model Zoob [38], and the MobileNet_SSD_v1 pre-trained model on the MS COCO dataset [38] to detect a Face. Training can be done on the computer or remotely like AWS, Google Cloud, paperspace, etc. The model was trained until acceptable accuracy.
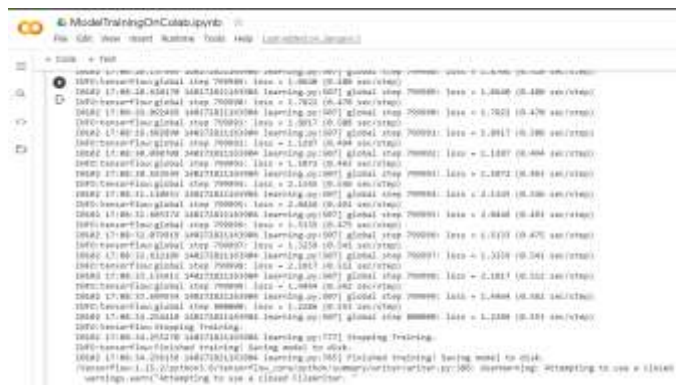

Fig. 7. Dataset Training on Colab.

After finishing with training, we exported the trained model to a single file Tensorflow graph. It is called a frozen inference graph. Fig. 7 presents the dataset training process in colab. All the dataset and the code are available at https://github.com/SshShamma/Drowsiness_Detection_DeepLearning.

VII. RESULT AND DISCUSSION

In this section, the experiment results and associate outcomes have been explained in details.

A. Experiment Results

The experiment is performed in Google cloud using Tensorflow Object Detection API. We train our SSD model with an average step speed of 0.5 s, which means that training

one network for 800 k steps takes around six days. After completing our training, we get a frozen inference graph, and then we transform this graph into an inference model. Our inference model was run on a laptop for initial testing and gave 8 frames per second (FPS) for SSD architecture.

PASCAL VOC is a popular dataset and an evaluation tool for object detection and classification. The trained model MobileNet_SSD_v1 is tested using the PASCAL VOC assessment metric on the test dataset. The intersection over Union (IoU) is defined as the area at which the projected bounding box (B) and the ground-truth box (Bgt) are intersected, separated by their union area.

$$IoU = \frac{Area(B \cap Bgt)}{Area(B \cup Bgt)}$$

We consider that, depending on three factors, the detection is a true positive (TP) or false positive (FP). The first one is if the confidence score is higher than 0.5, the second one is the IoU of the predicted bounding box is higher than the ground-truth IoU threshold, and the third one is the predicted class corresponds to the ground truth class. The identification is false positive (FP) if the last two conditions are not fulfilled.

If the same ground-truth applies to multiple forecasts, then the one with the highest confidence score counts as a true positive and the others count as false positives. If the detection confidence level intended to detect a ground-truth is lower than the confidence level, the detection is counted as a false negative (FN). Here are the equations for precision, recall based on assumptions.

$$Precision = \frac{TP}{TP+FP}$$

The number of true positives divided by the sum of true positives and false positives is Precision. Average precision (AP) is used to test the efficiency of the object detectors. For our dataset, AP is calculated based on our four classes. Using the below formula, we get the mean average accuracy (mAP).

$$mAP = \sum_{i=1}^{k} \frac{APi}{k}$$

Here, AP is Average Precision and k is several ground truth boxes.

Table I shows the summary of the result.

TABLE I. RESULT SUMMARY

AP eyes close	AP eyes open	AP No Yawn	AP Yawn	mAP
0.962631	0.977191	0.997952	0.983024	0.980199

Fig. 8 shows the Loss function graph from which we can identify minimal losses that occurred.

From Fig. 9, we can see the confidence scores, with bounding boxes representing the object detections produced by the model for a given class at a specific moment in time.
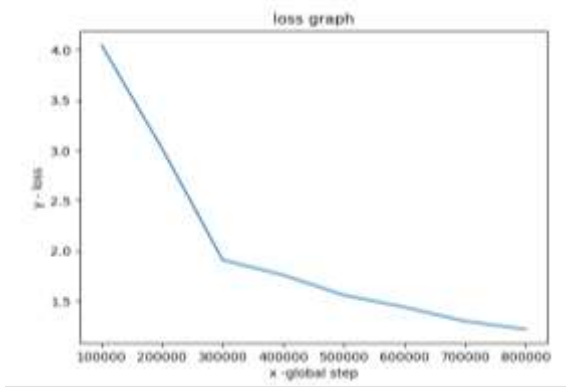
Fig. 8.   Loss Function Graph.

Fig. 9.   Trained Model in Action.

B. Discussion

In the previous work machine learning algorithm was used for manual feature extraction but nowadays deep learning architectures remove the manual feature extraction. It does automatic learning. Now deep learning architectures extract its feature in an automated process. So, a lot of time is saved here otherwise we have to spend time on determining the feature requirements to improve the classification result.

The dataset of the drowsiness observation model was largely tested with several light conditions, under a broad range of conditions, subjection & obstruction. In normal cases, our model works fine. It gives more than 90% accuracy. We tested the model on different ages and colors of people and get a good result. The success rate while driving in real-world situations was considered flawless. We also found that its efficiency is not much good in some cases. In low-light conditions or when the flashlight is pointed to the camera lens in the background, at that time, the performance of the trained model was not fully satisfied. But when the dataset doesn't accommodate any low light images, it was appropriate. That is why we only require it to function well under this condition.

VIII. CONCLUSION AND LIMITATIONS

The paper described an improved drowsiness detection system based on CNN. The key goal is to ensure that a lightweight approach is applied in integrated devices while maintaining and achieving high efficiency. In the trained model, we only use 250 low-light images. The main improvement done in the future is adding more low-light photos to work well in low-light conditions. Further enhancement of the dataset is required using a yawning dataset

as we could not use those annotations for detecting drowsiness. Another improvement area is the SSD_MOBILENET architecture for a better suit for drowsiness detection.

REFERENCES

[1] Amodio,A.,Ermidoro,M.,Maggi,D.,Formentin,S., Savaresi, S.M.(2018). Automatic detection of driver impairment based on pupillary light reflex.IEEE Transactions on Intelligent Transportation Systems, pp. 1-11.

[2] Yang, J.H., Mao, Z.H., Tijerina, L., Pilutti, T., Coughlin, J.F., Feron, E. (2009).Detection of driver fatigue caused by sleep deprivation. IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans, 39(4):694-705.

[3] Hu, S., Zheng, G. (2009). Driver drowsiness detection with eyelid related parameters by support vector machine. Expert Systems with Applications, 36(4):7651-7658.

[4] K. Fujiwara, E. Abe, K. Kamata, C. Nakayama, Y. Suzuki, T. Yamakawa, T. Hiraoka, M. Kano, Y. Sumi, F. Masuda, M. Matsuo, and H. Kadotani, ''Heart rate variability-based driver drowsiness detection and its validation with EEG,'' IEEE Trans. Biomed. Eng., vol. 66, no. 6, pp. 1769–1778, Jun. 2019.

[5] C. F. P. George, ''Sleep apnea, alertness, and motor vehicle crashes,'' Amer. J. Respiratory Crit. Care Med., vol. 176, no. 10, pp. 954–956, Nov. 2007.

[6] J. A. Owens, T. Dearth-Wesley, A. N. Herman, and R. C. Whitaker, ''Drowsy driving, sleep duration, and chronotype in adolescents,'' J. Pediatrics, vol. 205, pp. 224–229, Feb. 2019.

[7] Mardi,Z.,Ashtiani,S.N.,Mikaili,M.(2011).   EEG-based   drowsiness detection for safe driving using chaotic features and statistical tests.Journal of Medical Signals And Sensors,1(2):130-137.

[8] M. Benz, "Mercedes benz safety - s class." [Online]. Available: https://www.mercedes-benz.co.uk/passengercars/mercedes-benz-cars/models/s-class/saloon-w22/safety/intelligent-drive.module.html.

[9] Muze, "Eyesight." [Online]. Available: https://www.eyesight-tech.com/.

[10] Jabbar, R., Shinoy, M., Kharbeche, M., Al-Khalifa, K., Krichen, M., & Barkaoui, K. (2020, February). Driver drowsiness detection model using convolutional neural networks techniques for android application. In *2020 IEEE International Conference on Informatics, IoT, and Enabling Technologies (ICIoT)* (pp. 237-242). IEEE.

[11] Noori,S.M.,Mikaeili,M.(2016).Driving drowsiness detection using a fusion of electroencephalography, electrooculography, and driving quality signals. J Med Signals Sens,6: 39-46.

[12] Using, S., Eog, E.: Development of Vehicle Driver Drowsiness Detection System Using Electrooculogram (EOG). In: 1st International Conference on Computers, Communications, and Signal Processing With Special Track on Biomedical Engineering (CCSP), Kuala Lumpur, Malaysia. pp. 165–168 (2005).

[13] S. K. L. Lal and A. Craig, ''A critical review of the psychophysiology of driver fatigue,'' Biol. Psychol., vol. 55, no. 3, pp. 173–194, Feb. 2001.

[14] C. C. Liu, S. G. Hosking, and M. G. LennØ, ''Predicting driver drowsiness using vehicle measures: Recent insights and future challenges,'' J. Saf. Res., vol. 40, pp. 239–245, Aug. 2009

[15] Q. Ji and X. Yang, ''Real-time eye, gaze, and face pose tracking for monitoring driver vigilance,'' Real-Time Imag., vol. 8, no. 5, pp. 357–377, 2002.

[16] Takei, Y., Furukawa, Y.: Estimate of driver's fatigue through steering motion. 2005 IEEE Int. Conf. Syst. Man Cybern. 2, 1–6 (2005).

[17] Lee, A.: Comparing Deep Neural Networks and Traditional Vision Algorithms in Mobile Robotics. Swart. Coll. (2015).

[18] Lowe, D.G. Distinctive Image Features from Scale-Invariant Keypoints. Int. J.Comput. Vis. (2004).

[19] Lazebnik, S., Schmid, C., Ponce, J.: Beyond bags of features: spatial pyramid matching for recognizing natural scene categories. IEEE Comput. Society.

[20] Dwivedi, K., Biswaranjan, K., Sethi, A (2014). Drowsy driver detection using representation learning.Advance Computing Conference(IACC), IEEE,pp. 995-999.

[21] Alshaquaqi,B.,Baquhaizel,A.S.,AmineOuis,M.E.,Boumehed,M.,Ouamri, A.,Keche,M.(2013).Driver drowsiness detection system. 8th International Workshop on Systems,Signal Processing and Their Applications(WoSSPA).

[22] Shakeel M.F., Bajwa N.A., Anwaar A.M., Sohail A., Khan A., Haroon-ur-Rashid (2019) Detecting Driver Drowsiness in Real Time Through Deep Learning Based Object Detection. In: Rojas I., Joya G., Catala A. (eds) Advances in Computational Intelligence. IWANN 2019. Lecture Notes in Computer Science, vol 11506. Springer, Cham. https://doi.org/10.1007/978-3-030-20521-8_24.

[23] 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.

[24] Said, S., AlKork, S., Beyrouthy, T., Hassan, M., Abdellatif, O., Abdraboo, M.F. (2018). Realtime eye-tracking and detection-a driving assistance system.Advances in Science, Technology and Engineering Systems Journal,3(6):446-454.

[25] Mandal,B.,Li,L.,Wang,G.S.,Lin,J.(2017). Towards detection of bus driver fatigue based on robust visual analysis of eye state. IEEE Transactions on Intelligent Transportation Systems, 18(3):545557.

[26] 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.

[27] Saini, V., & Saini, R. (2014). Driver drowsiness detection system and techniques: a review. *International Journal of Computer Science and Information Technologies*, *5*(3), 4245-4249.

[28] P. Peyrard, "Personal system for the detection of a risky situation and alert," Feb. 28, 2019, us Patent App. 16/178,365.

[29] Jabbar, R., Al-Khalifa, K., Kharbeche, M., Alhajyaseen, W., Jafari, M., Jiang, S. (2018). Real-time driver drowsiness detection for an android application using deep neural networks techniques. Procedia Computer Science,130:400-407.

[30] Arcaya-Jordan, A., Pegatoquet, A., & Castagnetti, A. (2019, March). Smart Connected Glasses for Drowsiness Detection: a System-Level Modeling Approach. In *2019 IEEE Sensors Applications Symposium (SAS)* (pp. 1-6). IEEE.

[31] Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H.: MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. (2017).

[32] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C., Berg, A.C.:SSD: Single Shot MultiBox Detector. In: European Conference on Computer Vision. pp. 21–37 (2015).

[33] Huang, J., Rathod, V., Sun, C., Zhu, M., Korattikara, A., Fathi, A., Fischer, I., Wojna, Z., Song, Y., Guadarrama, S., Murphy, K.: Speed/accuracy trade-offs for modern convolutional object detectors. Proc. - 30th IEEE Conf. Comput.Vis. Pattern Recognition, CVPR 2017. 2017–Janua, 3296–3305 (2017).

[34] Song, F., Tan, X., Liu, S.: Eyes Closeness Detection from Still Images with Multi-scale Histograms of Principal Oriented Gradients.

[35] Pan, S.J., Fellow, Q.Y.: A Survey on Transfer Learning. IEEE Trans. Knowl. Data Eng. 22, 1345–1359 (2010).

[36] Oquab, M., Bottou, L., Laptev, I., Sivic, J.: Learning and Transferring Mid-Level Image Representations using Convolutional Neural Networks. IEEE Conf. Comput. Vis. Pattern Recognit. 1717–1724 (2014).

[37] Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.:Microsoft COCO: Common objects in context. Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics). 8693 LNCS, 740–755 (2014).

[38] Huang, J., Rathod, V., Sun, C., Zhu, M., Korattikara, A., Fathi, A., Fischer, I., Wojna, Z., Song, Y., Guadarrama, S., Murphy, K.: Speed/accuracy trade-offs for modern convolutional object detectors. Proc. - 30th IEEE Conf. Comput.Vis. Pattern Recognition, CVPR 2017.2017–Janua, 3296–3305 (2017).

[39] Sahayadhas, A., Sundaraj, K., & Murugappan, M. (2012). Detecting driver drowsiness based on sensors: a review. *Sensors*, *12*(12), 16937-16953.

[40] Albawi, S., Mohammed, T. A., & Al-Zawi, S. (2017, August). Understanding of a convolutional neural network. In *2017 International Conference on Engineering and Technology (ICET)* (pp. 1-6). Ieee.

[41] Jana, A. P., & Biswas, A. (2018, May). YOLO based Detection and Classification of Objects in video records. In *2018 3rd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT)* (pp. 2448-2452). IEEE.

[42] Chen, C., Liu, M. Y., Tuzel, O., & Xiao, J. (2016, November). R-CNN for small object detection. In *Asian conference on computer vision* (pp. 214-230). Springer, Cham.

[43] Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, *28*, 91-99.