U. Vara Prasad, 2024, 12:3
ISSN (Online): 2348-4098
ISSN (Print): 2395-4752

International Journal of Science,
Engineering and Technology

An Open Access Journal

Intelligent Edge-Based Driver Fatigue Identification in Mobile Crowdsourcing

Assistant Professor U. Vara Prasad, Assistant Professor B. Srinivas, K. Bharath Nag, J. Anusha, S. Sushma, V. Sunil Kumar, S. Ravindhar Reddy

Computer science and Engineering

Bomma institute of technology and science Khammam, Telangana, India.

Abstract- Drowsy drivers pose a serious risk to public safety through their involvement in traffic accidents. According to recent data, drunk drivers are thought to be responsible for 15.5% of fatal collisions. A sleepiness detection system can help prevent these incidents a great deal, especially with the increasing usage of roadside units and mobile devices. Although a number of solutions have been put out in the literature, none of them fully presents a distributed architecture that can satisfy these applications' requirements without invading the privacy of the drivers. In this research, a smart edge computing-based two-stage driver drowsiness detection system is proposed. Without disclosing their personal information, mobile devices in the vehicle are utilized to record and evaluate the drivers' present states. When sleepiness is verified, the smart edge is used as a decision-maker.

Keywords- Convolutional neural network (CNN), deep learning, smart edge and drowsiness detection.

I. INTRODUCTION

About 328,000 accidents are caused by sleepy driving annually [1]. It is involved in around 15.5% and 13.1%, respectively, of fatal incidents that result in injuries and fatalities [2]. It occurs when a driver is extremely sleepy, rendering him unable to react to traffic situations while operating a vehicle. Reduced driving accidents due to sleepiness might be achieved with the use of a real-time, precise system for detecting sleepiness. However, the shortcomings of the approaches that current solutions employ place limitations on them. Three primary procedures are employed in the literature to identify the state of drivers: computer vision approaches, physiological-based methodologies, and vehicle behavior monitoring methods [3]. In order to construct detection systems, these approaches rely on several properties. Methods of vehicle behaviour Make use of patterns like steering activity and variations in the vehicle's location with respect to road characteristics [4]. These techniques, however, are less dependable than the other strategies due to the unpredictable nature of the driving environment and variations in driving styles across drivers. Wearable device-collected heart rate variability and electroencephalogram (EEG) measurements are essential to physiologically-based approaches. Nevertheless, vibrations from engines and motion can affect EEG equipment. As a result, they perform poorly in actual driving situations [5]. Lastly, head orientation, yawning, and eye movement derived from video feeds and photos are used in computer vision algorithms. ECG readings. The RGB colors that make up the UTA-RLDD dataset videos gathered from individuals of different ethnicities. The UTA-RLDD dataset differs from the other dataset mostly in because the dataset records the participants as they drift off to sleep, therefore capturing the subtle changes in expression that occur as the participants get sleepy.

A deep drowsiness detection (DDD) network is proposed in [7]. Features are extracted from both RGB videos and optical flow, and then the features are passed to three different networks: Flow

ImageNet extracts head and facial movements, VGG-Face Net learns features related to drowsiness but is more sensitive to appearance variations, and AlexNet extracts features related to drowsiness. On the NTHU dataset, this architecture achieves 73.06% detection accuracy.

A more tailored strategy for buses and heavy trucks is suggested in [8]. This idea makes predictions based on the driver's entire upper body and facial characteristics using the dome cameras that are currently in place. It takes into account posture variations using a multi-model method.

In contrast, machine learning algorithms demonstrated competitive outcomes when it came to the task of classifying driving behavior based on smartphone usage. The authors of [10] used feature extraction, feature selection, and classification techniques to separate safe driving practices from aggressive driving styles. They extracted 78 driving features from preprocessed 3-axis accelerometer sensor readings, which were divided into five sets in the time and frequency domains: jerk profile feature sets in the time domain, correlation coefficient, data threshold violation, and histogram feature sets. Six of the 78 characteristics were chosen using a random forest classifier for the classification test, and they had a 95.5% classification accuracy. The study [11] examined the performance of many sensors, including the accelerometer sensor, using data from

II. METHODOLOGY

As previously stated, the method is based on two independent steps: the driver side does the local sleepiness detection step, and the edge side performs the drowsiness confirmation step.

1. Model for Local Drowsiness Detection

This section outlines the suggested vision-based sleepiness detection techniques, which use input pictures collected from a video stream that uses an agent-integrated front-facing camera to capture the driver's face and generate a binary classification of the driver's tiredness. A face-based sleepiness detector that uses the entire face as input to produce the categorization is the first tested method. The eye and mouth regions of interest (ROIs) are used in the second tested method. This method builds two classifiers, the outputs of which are combined to determine whether or not the driver is sleepy. The mouth ROI is classified into the normal or yawing condition by the second classifier, while the eye ROIs are classified into open and closed classes by the first classifier. When to determine whether or not the driver is sleepy. The mouth ROI is classified into the normal or yawing condition by the second classifier, while the eye ROIs are classified into open and closed classes by the first classifier. A motorist is deemed tired if their mouth is yawning and their eye is closed, but they appear normal elsewhere.

Using a Face-based Approach to Detect Sleepiness

Face characteristics that are taken from the entire face are used in the first method of sleepiness detection. The suggested model utilizes Five frames per second (FPS) of picture frames taken from movies are put into the model to train it in two different scenarios. employing raw frame photos from a single scenario, the model is trained.

Without employing face identification algorithms to trim the face area of interest (ROI). In order to trim the face ROI from the frames and provide the data to the model, the second scenario looks at face detection approaches. Figure 1 displays samples of the input data for each of the two scenarios. The face region is found using the built-in dlib face detector, which is then cropped from the picture and sent into the model.

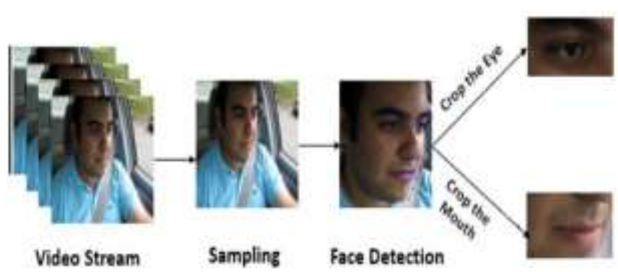


Fig.1. Input Data Samples for the Face-based Drowsiness Detection Model.

A pretrained VGG16 model that has previously been trained on the ImageNet dataset is used to construct the face-based sleepiness detector. The final fully connected (FC) layer of the VGG16 model is fine-tuned using the extracted frames from the movies in the NTHU dataset while the pre-trained convolutional layers of the model are first frozen. After the fully connected layer and L2 regularizes were added to the FC and output layers, a dropout layer was added to prevent the model from overfitting. Because the pre-trained VGG16 network layers were frozen, the number of trainable parameters decreased from 15,894,849 to just 1,180,161 parameters. This substantially shortened the training time and made advantage of the deep VGG16 architecture.

A ROI-Based Eye and Mouth Drowsiness Detection Approach

An additional method that makes use of facial traits connected to the eye and mouth region of interest is suggested in order to enhance the outcomes of the driver sleepiness detection implemented on the worker nodes.

The movies that were recorded for the driver are first pre-processed using a script that extracts picture frames at a rate of five frames per second. Next, the face and eye ROIs are cut from the picture and sent to the mouth and eye classifiers in tandem. To lower the computational detection, just the eye that is closer to the camera is taken into account.

Period without impacting the effectiveness of the model [10]. The eye classifier receives the ROI of the eye as input and produces one of two classes: closed or open eyes. The face ROI is the mouth classifier's input, while the normal classes and yawn are its outputs.

The two classifiers' frame output is used to establish the driver's level of tiredness. If the driver is yawning and has one eye closed, he is deemed sleepy; if not, he is deemed normal. The CNN model is utilized to create the eye classifier, and figure 2 displays an example of the input data that was used to train the classifier. The suggested architecture was developed by looking at several network modifications.

(a) Closed Eye (b) Open Eye

Fig.2. Input Data Samples for the Eye ROI Classification Model.

2. Architectural Diversity

The processes for establishing the framework architecture as they are shown in algorithm 1 are explained in this part. First, as seen in figure 2, the CCTVs placed in streets, intersections, and traffic signals are referred to as edge nodes. Because they must constantly retrieve data from sleepy drivers, roadside CCTVs—which can be uni- or multi-directional—are regarded as high-performance devices.

Algorithm 1 LEN and MEN Selection, Clustering, and Drowsiness Detection

```
Input : , Participant's coordinate, Connectivity_type = Wireless, Bluetooth,
        or CCTV, Accelerometers reading
Output: LENs, MEN, Drowsiness
initialization
for every connected participant do
    Check the shared participant's connectivity type
    if W_i connectivity_type = CCTV then
        Add W_i to the list of preselected EN else
            W_i is driver
        end
    end
    end

    for Every EN in predefined list do
        calculate EN's objective function
        Objective_function = w_1 × RE_i^W + w_2 × U_i^W + w_3 × Accuracy

        for every time units do
            Calculate the number of optimal clusters as
            best_ size = list of LENs - 1
            if LEN_ij = Maxt_objective function then
                LEN_ij is MEN_ij
            FinalLENs = drop MEN_ij from list of LEN_ij
            Construct the clusters as:
            Cr_id = kmeans(L_i^W , best_size)
            Centroids = best_size LEN_ij s
            for Every Cluster do
                Select W_g^Cr where driver coordinate are within CCTV
                range
                Detect driver car
                Calculate car accelerators
                Fuse local detection with accelerometers readings
                Apply LSTM Algorithm
                Confirm Drowsiness
            end
        end
```

The distance between every edge node is 50 meters which is the detection range of CCTV (lines 2-9). Once the list of LENs (playing the role of CCTVs) is pre-defined, an objective function is calculated using LENs computing capabilities in term of energy consumption REWi and computing unit UW and devices accuracy (lines 11-13) to define a final list of LENs and then the MEN. The MEN is LEN with the highest objective function, responsible for reporting the overall detection to authorities to stop the drowsy driver (lines 16-20). To cope with the mobility of cars and the change in the CCTV range, the clustering process is dynamic and relies on LENs and car locations (X and Y Coordinates).

Each re-clustering is the start of a new cycle. The cycle ends after the MENs confirmation of drowsiness detection. The length of each cycle is set to five-time units. A relatively small process is selected to track the drivers as they are moving continuously.

fter each cycle, LENs share records of drivers with sufficient data to the MEN, and the rest of the data is forwarded to the other LENs as carry-over. At the beginning of the next cycle, each LEN keeps only the carry-over records of its detected drivers and continues accumulating data. If a driver is detected as drowsy, his data collection continues until he gets a MEN drowsiness confirmation. This feature enables the solution to keep track of malicious drivers over time (lines 25-31).

3. Classification of Driving Actions Confirmation of Drowsiness

The mobile client records a video feed, which is used to identify driver tiredness and initiate the drowsiness detection procedure. The LENs gets the detection result and proceed to monitor the sleepy drivers in order to gather more accelerometer readings and drowsiness detection data. The MEN receive the LENs' detection records for each motorist who exhibits signs of sleepiness after they have generated enough acceleration measures to support the LENs' findings. The result is a list of individuals who have been verified to be sleepy. A methodology for classifying driving behavior is used on the MEN side. It detects driving behavior over a sliding window of data by using the acceleration records of the drivers that are gathered over time. To verify the Considering the commonalities between the classes, the following five are included abrupt right, left swerving, abrupt acceleration, and breaking. As indicated in table 5, a range of distinct sequence lengths are employed to assess the categorization accuracy.

III. SIMULATION PARAMETERS

Four datasets are taken into consideration in this work: the Sarwat Foursquare databases [11], the NTHU dataset [16], and the Kaggle sleepiness detection dataset [20]. Two subsets of the Kaggle dataset are available for classification: the first classifies the eye ROI into closed eye and open eye, while the second classifies the mouth as either normal or yawning. There are 2900 640 × 480-pixel photos in the entire collection. On the other hand, films of drivers in various daytime and nighttime lighting conditions may be found in the NTHU collection. The AVI format videos were gathered with a resolution of 640 × 480 pixels. Using an infrared (IR) lamp, low-light (nighttime) video footage is captured. It includes training and assessment films featuring eighteen subjects. Techniques for enhancing data are utilized. shear effects, rescaling, and horizontal flipping methods to both datasets. The eye ROIs are shrunk to 64 × 64 pixels, while the face ROI pictures are down-sampled to 112 × 112 pixels in order to simplify the method.

The model's acceleration dataset may be found in [19]. Acceleration readings for events along the x, y, and z axes are included. The action segments are from 38 and 200-time steps long. Sequences that are utilized to actively choose whether to break or accelerate are divided into four categories: regular behavior, sudden breaking, sudden acceleration, and sudden swerving. At the same time, if the driver lets off of the steering wheel, the car may veer without warning. While quick acceleration and sudden breaking are deemed natural, abrupt swerving is seen as drowsiness. In this model, data regarding the driver’s gadgets, including energy, sensor availability, and device accuracy, are obtained from the Sarwat Foursquare dataset [11], a dataset for social networking apps. Table 1 lists additional simulation settings for the sleepiness detection model.

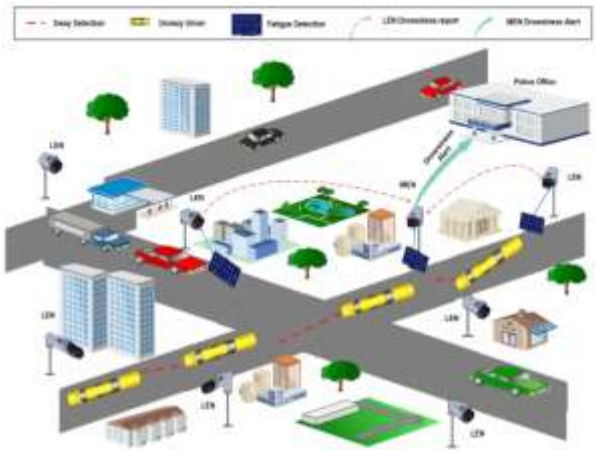


Fig.3. Edge-based distributed architecture for Drowsiness Detection.

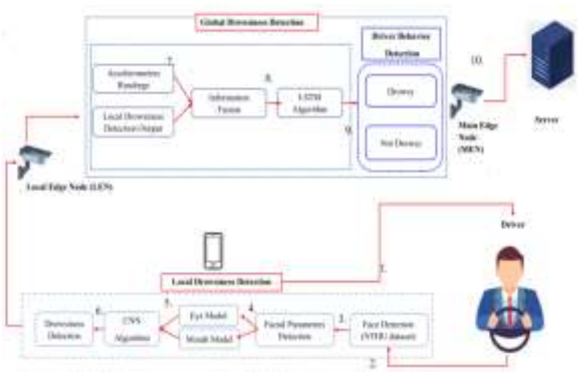


Fig.4. Conceptual Architecture.

IV. EVALUATION PARAMETERS

The main performance metrics are described as follows

1. Precision Rate

$$Precision = \frac{TP}{TP + FP} * 100\% \qquad (1)$$

where FP stands for false positive and TP for true positive. The percentage of samples expected to be genuine positives in the sample of positive instances is shown by the precision rate.

2. Recall Rate

$$Recall = \frac{TP}{TP + FN} * 100\% \qquad (2)$$

The recall rate shows the percentage of samples that were anticipated to be negative cases out of all positive samples, and FN stands for false negatives.

3. F1-score

$$F1 - score = \frac{2 * Precision * Recall}{Precision + Recall} \qquad (3)$$

4. Accuracy

It measures how accurate the model is in providing correct predictions.

$$Accuracy = \frac{Number\ of\ correct\ predictions}{Total\ number\ of\ predictions} \qquad (4)$$

Table 1. Implementation Parameters.

Parameters	Description
Datasets	NTHU, DoTA, OffSEC datasets
Driver Parameters	
Number of Participants	500
Number of edge nodes	4
Drivers Location (Lat, Long)	([31....43], [129....144])
Drivers Information	Video Streaming, Accelerometers readings
Connectivity type	Wireless, Bluetooth, CCTV
Car Information	ID, Name
Sensors Type	Residual Energy, CPU, Accuracy
Weather Conditions	Sunny, Snowy, Rainy
Implementation Parameters	
Algorithms	CNN, LSTM
Train_Val_Test Split	80%, 20%
Number of Epochs	30 (ROI-based), 10 (Face-based)
Optimizers	Adam (Eye Model & Face-based), RMS prop (Mouth Model)
Learning Rate	0.0005 (ROI-based & Face-based)

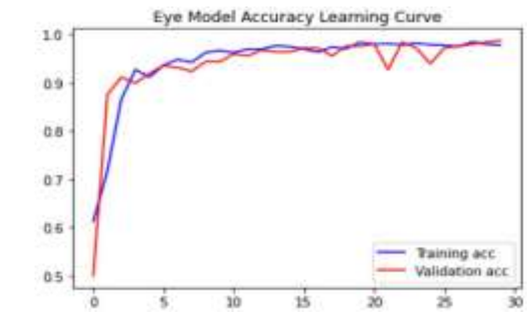
Face-Based Drowsiness Detection Approach

The 16 patients in the NTHU dataset comprised the training dataset that was utilized to train the suggested face-based sleepiness detection algorithm. 20% of the training set is chosen at random to validate the model, and a testing subset of the NTHU dataset is used for testing. With 256 neurons in the FC layer and learning rates of 0.0005 for the pre-processed face frames scenario and 0.0005 for the RMSprop optimizers, respectively, the best detection results are obtained. A batch size of 64 and 10 epochs were used to train both models.

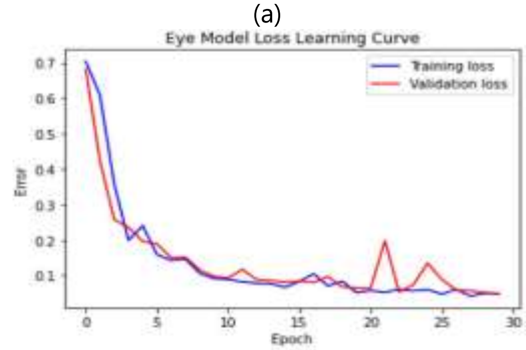
Eye And Mouth Classifier

Once the model is trained and validated on the NTHU and Kaggle datasets, it is tested on the training set of the Kaggle eye and mouth dataset [20], which is randomly divided into training and validation subsets." To get the best classification results, train the model using Adam optimizer for eye training and RMSprop optimizer for mouth training across 30 epochs with 16 batches at a learning rate of 0.0005. Using the Kaggle dataset rather than the NTHU dataset, it is evident that the highest classification accuracy is obtained for both the mouth and the eye. Eye ROI and Mouth ROI are two subgroups of the Kaggle dataset that, as previously mentioned, help the algorithm learn and lower the loss function. On the other hand, because the NTHU dataset's learning process is created from scratch, processing the films will take longer and require more powerful computers, which might result in more loss.

The accuracy performances throughout the training and validation stages are shown in Figures 5(a) and 6(a). As demonstrated, CNN quickly improved training and validation accuracy throughout the course of the epochs. This indicates that the model is well-trained as it achieved an ideal accuracy of 98% and 97% for mouth and eye identification, respectively. Nonetheless, figures 5 (b) and 6 (b) demonstrate the performance of the loss function. The CNN model's loss function was reduced during training and validation in order to get the lowest feasible value. For mouth and eye detection, this resulted in convergent values of 5% and 10% after 30 epochs.

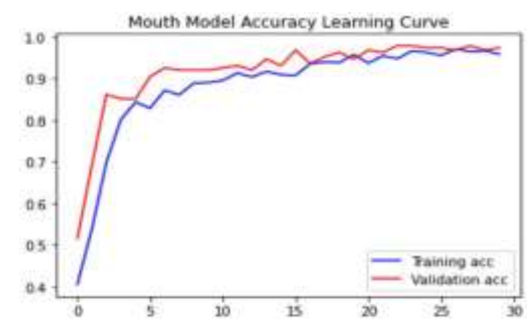


(a)

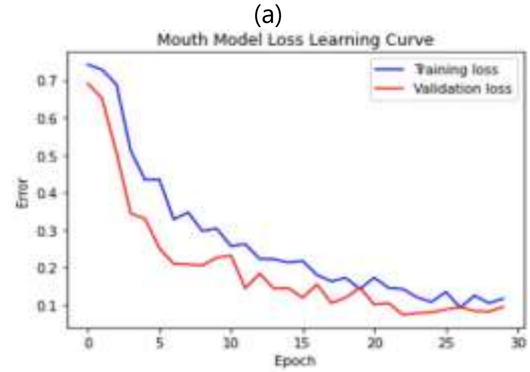


(b)

Fig.5. Training vs. Validation Accuracy and Loss for Eye Detection.



(a)



(b)

Fig.6. Training vs. Validation Accuracy and Loss for Mouth Detection.

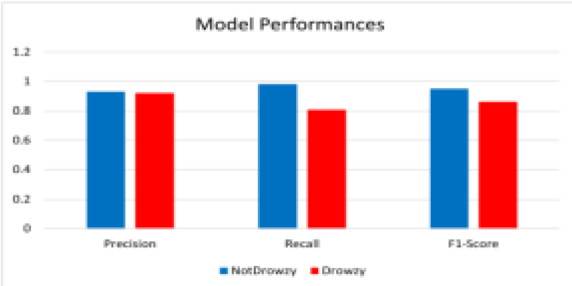


Fig.7. Prediction Performances of Driving Behaviour Classification.

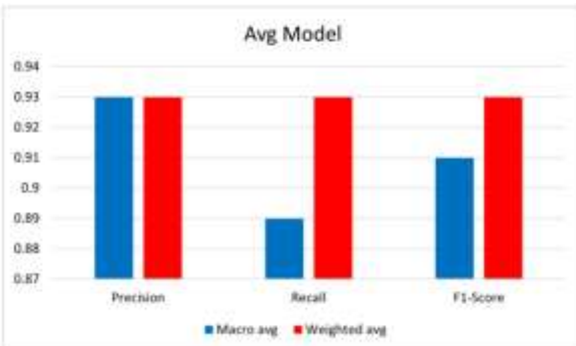


Fig.8. Average metrics of Driving Behaviour Classification.

The sequence with a length of 16 and the fewest records yielded the highest accuracy for the trained model, 93%. In terms of accuracy, recall, and F1-score, LSTM performed well for the final identification of sleepiness. According to Figure 7, the model's detection precision for the not-drowsy and drowsy classes is 93% and 92%, respectively. In a similar vein, the awake classes outperformed the sleepy class, achieving high memory and F1-score of 98% and 95%, respectively, compared to 81% and 86% for recall and F1-score, respectively. Regardless of the amount of each category in the dataset, Figure 8 shows the macro-average and weighted-average used to calculate precision, recall, and F1-Score for each. Both metrics achieved similar precision of 93% while they are different for recall and F1-score where macro-avg is lower than weighted-avg.

V. CONCLUSION AND FUTURE WORK

This research suggests a sleepiness detection system that can identify drowsiness accurately. It can overcome the problems associated with deploying critical systems on centralized architectures since it is deployed on a distributed architecture. Two stages of detection are used in the implementation of the sleepiness detection system: local detection via facial expression and global detection via the combination of local and driving behavior detections. CNN models yield 97.3% and 98.2% accuracy for mouth and eye classifiers, respectively. The results of the two classifiers and the accelerometer measurements from the automobile are used to calculate the overall sleepiness state. The driving behavior categorization model verified that the 93% accurate LSTM algorithm is used at the edge level for driver sleepiness detection. In future studies, additional factors like heart rate and sensor body readings will be taken into account to verify the driver's tiredness.

REFERENCES

1. Fatigued Driver National Safety Council. Accessed. 2022. [Online]. Available: https://www.nsc.org/road/safety-topics/fatigued-driver.
2. (2010). The Prevalence and Impact of Drowsy Driving. [Online]. Available: https://aaafoundation.org/prevalence-impact-drowsy-driving/.
3. M. K. Hussein, T. M. Salman, A. H. Miry, and M. A. Subhi, ''Driver drowsiness detection techniques: A survey,'' in Proc. 1st Babylon Int. Conf. Inf. Technol. Sci. (BICITS), Apr. 2021, pp. 45–51.
4. S. Lawoyin, D.-Y. Fei, and O. Bai, ''Accelerometer-based steering-wheel movement monitoring for drowsy-driving detection,'' Proc. Inst. Mech. Eng., D, J. Automobile Eng., vol. 229, no. 2, pp. 163–173, 2015.
5. H. Iwamoto, K. Hori, K. Fujiwara, and M. Kano, ''Real-driving implementable drowsy driving detection method using heart rate variability based on long short-term memory and autoencoder,'' IFAC-Papers OnLine, vol. 54, no. 15, pp. 526–531, 2021.
6. S. Khare, S. Palakkal, T. V. Hari Krishnan, C. Seo, Y. Kim, S. Yun, and S. Parameswaran, ''Real-time

driver drowsiness detection using deep learning and heterogeneous computing on embedded system,'' in Computer Vision and Image Processing (Communications in Computer and Information Science), N. Nain, S. K. Vipparthi, and B. Raman, Eds. Cham, Switzerland: Springer, 2020, pp. 86–97.
7. S. Park, F. Pan, S. Kang, and D. C. Yoo, ''Driver drowsiness detection system based on feature representation learning using various deep networks,'' in Proc. Asian Conf. Comput. Vis. in Lecture Notes in Computer Science, C.-S. Chen, J. Lu, and K.-K. Ma, Eds. Cham, Switzerland: Springer, 2017, pp. 154–164.
8. B. Mandal, L. Li, G. S. Wang, and J. Lin, ''Towards detection of bus driver fatigue based on robust visual analysis of eye state,'' IEEE Trans. Intell. Transp. Syst., vol. 18, no. 3, pp. 545–557, Mar. 2017.
9. J. Lyu, Z. Yuan, and D. Chen, ''Long-term multi-granularity deep framework for driver drowsiness detection,'' CoRR, vol. abs/1801.02325, 2018.
10. M. Hashemi, B. Farahani, and F. Firouzi, ''towards safer roads: A deep learning-based multimodal fatigue monitoring system,'' in Proc. Int. Conf. Omni-Layer Intell. Syst. (COINS), Aug. 2020, pp. 1–8.
11. H. Lamaazi, R. Mizouni, S. Singh, and H. Otrok, ''A mobile edge-based Crowd Sensing framework for heterogeneous IoT,'' IEEE Access, vol. 8, pp. 207524–207536, 2020.
12. H. Lamaazi, R. Mizouni, H. Otrok, S. Singh, and E. Damiani, ''Smart 3DM: Data-driven decision making using smart edge computing in hetero-crowdsensing environment,'' Future Gener. Comput. Syst., vol. 131, pp. 151–165, Jun. 2022.
13. J. W. Baek, B.-G. Han, K.-J. Kim, Y.-S. Chung, and S.-I. Lee, ''Real time drowsiness detection algorithm for driver state monitoring systems,'' in Proc. 10th Int. Conf. Ubiquitous Future Netw. (ICUFN), Jul. 2018, pp. 73–75.
14. 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 Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW), Jul. 2017, pp. 438–445.
15. S. Abtahi, M. Omid Yeganeh, S. Shi Mohammadi, and B. Hariri, ''YawDD: A yawning detection dataset,'' in Proc. 5th ACM Multimedia Syst. Conf., Mar. 2014, pp. 24–28.
16. C.-H. Weng, Y.-H. Lai, and S.-H. Lai, ''Driver drowsiness detection via a hierarchical temporal deep belief network,'' in Proc. Asian Conf. Comput. Vis. in Lecture Notes in Computer Science, C.-S. Chen, J. Lu, and K.-K. Ma, Eds. Cham, Switzerland: Springer, 2017, pp. 117–133.
17. Q. Massoz, T. Langohr, C. Francois, and J. G. Verly, ''The ULg multimodality drowsiness database (called DROZY) and examples of use,'' in Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV), Mar. 2016, pp. 1–7.
18. R. Ghoddoosian, M. Galib, and V. Athitsos, ''A realistic dataset and baseline temporal model for early drowsiness detection,'' in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW), Jun. 2019, pp. 178–187.
19. M. R. Carlos, L. C. González, J. Wahlström, G. Ramírez, F. Martínez, and G. Runger, ''How smartphone accelerometers reveal aggressive driving behavior? —The key is the representation,'' IEEE Trans. Intell. Transp. Syst., vol. 21, no. 8, pp. 3377–3387, Aug. 2020.
20. G. Žylius, ''Investigation of route-independent aggressive and safe driving features obtained from accelerometer signals,'' IEEE Intell. Transp. Syst. Mag., vol. 9, no. 2, pp. 103–113, Apr. 2017.