TELKOMNIKA Telecommunication, Computing, Electronics and Control
Vol. 23, No. 3, June 2025, pp. 682∼693
ISSN: 1693-6930, DOI: 10.12928/TELKOMNIKA.v23i3.26059

Convolutional neural network-based real-time drowsy driver detection for accident prevention

Nippon Datta[1], Tanjim Mahmud[2], Manoara Begum[3], Mohammad Tarek Aziz[1], Dilshad Islam[4], Md. Faisal Bin Abdul Aziz[5], Khudaybergen Kochkarov[6], Temur Eshchanov[7], Valisher Sapayev Odilbek Uglu[8], Sobir Parmanov[9], Mohammad Shahadat Hossain[10,11], Karl Andersson[11]

[1]Department of Computer Science and Engineering, Chittagong University of Engineering and Technology, Chittagong, Bangladesh
[2]Department of Computer Science and Engineering, Rangamati Science and Technology University, Rangamati, Bangladesh
[3]Department of Computer Science and Engineering, Port City International University, Chittagong, Bangladesh
[4]Department of Physical and Mathematical Sciences, Chattogram Veterinary and Animal Sciences University, Chittagong, Bangladesh
[5]Department of Computer Science and Engineering, Comilla University, Comilla, Bangladesh
[6]Department of Special Science, Tashkent State University of Economy, Tashkent, Uzbekistan
[7]Urgech State University Named After Abu Rayhon Beruni, Urgench, Uzbekistan
[8]Department of General Professional Subjects, Mamun University, Khiva, Uzbekistan
[9]National University of Uzbekistan, Tashkent, Uzbekistan
[10]Department of Computer Science and Engineering, University of Chittagong, Chittagong, Bangladesh
[11]Cybersecurity Laboratory, Luleå University of Technology, Skellefteå, Sweden

Article Info

Article history:

Received Feb 7, 2024
Revised Feb 27, 2025
Accepted Mar 11, 2025

Keywords:

Convolutional neural network
Deep learning
Driver
Drowsiness
Lightweight convolutional neural network

ABSTRACT

Drowsy driving significantly threatens road safety, contributing to many accidents globally. This paper presents a convolutional neural network (CNN)-based real-time drowsy driver detection system aimed at preventing such accidents, particularly for deployment in Android applications. We propose a lightweight CNN architecture that effectively identifies drowsiness and microsleep episodes by categorizing driver facial expressions into four distinct categories: close-eye expressions, open-eye expressions, yawns, and no yawns. Our model, which employs facial landmark detection and various pre-processing techniques to enhance accuracy, achieves an impressive 96.6% accuracy. This performance surpasses several popular CNN architectures, including VGG16, VGG19, MobileNetV2, ResNet50, and DenseNet121. Notably, our proposed model is highly efficient, with only 0.4 million parameters and a memory requirement of 1.51 MB, making it ideal for real-time applications. The comparative analysis highlights the superior balance between accuracy and resource efficiency of our model, demonstrating its potential for practical deployment in reducing accidents caused by driver fatigue.

Corresponding Author:

Tanjim Mahmud
Department of Computer Science and Engineering, Rangamati Science and Technology University
Rangamati-4500, Bangladesh
Email: tanjim_cse@yahoo.com

1. INTRODUCTION

Driving while drowsy poses a significant risk to road safety, contributing to a rising number of traffic accidents globally [1]. The prevalence of these accidents underscores the critical need for effective drowsy driver detection systems to prevent potential fatalities and injuries. Tiredness while driving can result from various factors, including sleep deprivation, medication, alcohol consumption, or night shifts [2]. Statistics reveal the alarming frequency of traffic accidents caused by drowsy driving, making it a pressing issue in road safety. Injuries and fatalities resulting from car accidents rank among the leading causes of death worldwide, with a staggering number of casualties reported annually [3]. Detecting driver sleepiness is imperative for the development of automobile safety technologies aimed at preventing road accidents. The advancement of such technologies is crucial in light of the escalating number of traffic accidents attributed to drowsy driving [4]-[6].

Numerous studies have investigated various methods for detecting driver drowsiness to improve road safety. For instance, Knapik and Cyganek [7] used thermal imaging to detect yawning with 71% and 87% accuracy for cold and hot voxels, respectively. Kiashari *et al.* [8] monitored respiration patterns via facial thermal imaging, achieving 83% accuracy with k-nearest neighbor (KNN) and 92% sensitivity with support vector machine (SVM). Dalal and Triggs [9], a 3D-deep convolutional neural network (CNN) framework achieved 76.2% accuracy in monitoring eye, mouth, and head movements on the NTHUDDD dataset. Moujahid *et al.* [10] proposed a system using eye, head, and mouth movements, achieving 79.84% accuracy with a non-linear SVM. You *et al.* [11] introduced a 3D CNN-based method that reached 73.9% accuracy by analyzing facial features without pre-specification. Eye-tracking features were central to [12], with Random Forest achieving up to 91.18% accuracy. Mardi *et al.* [13] used electroencephalogram (EEG) data and neural networks for 83.3% accuracy in sleepiness classification. Noori *et al.* [14] combined EEG, Electrooculography, and driving signals for 76.51% accuracy using a self-organized map network. Other methods include steering pattern analysis [15], eye blink rate detection [16], yawning detection [17], CNN-based approaches [18], and advanced driver assistance systems [19].

This paper presents a CNN-based real-time drowsy driver detection system designed to mitigate the risks associated with tired driving. The proposed model leverages facial detection techniques and eye area identification to monitor driver fatigue. By utilizing advanced image processing algorithms [20], [21], the system detects key indicators of drowsiness, such as closed-eye expressions, open-eye expressions, and yawns.

The contributions of this paper include: i) introduce a lightweight CNN architecture for real-time drowsiness detection; ii) apply facial landmark detector and various pre-processing techniques to the dataset to identify major facial features; iii) comparison with other model memory requirements and parameters; and iv) comparison with other existing research work.

The subsequent sections of the paper are organized as follows: in section 2, we introduce the proposed methodology for drowsy driver detection. The results and analysis of the detection system are discussed in section 3, while section 4 encapsulates the paper's conclusion and outlines future directions for research in this domain.

2. METHOD

The fundamental structure of the suggested system is based on four modules: dataset description, pre-processing, lightweight CNN, and classification. The primary architecture of the proposed system is shown in Figure 1.

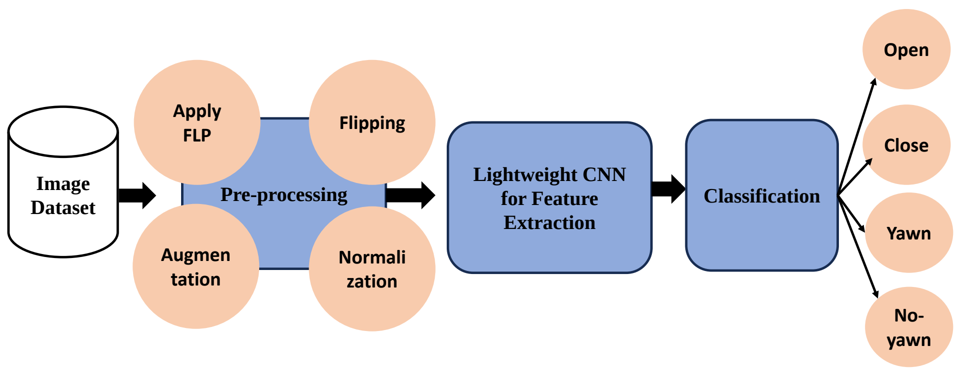

Figure 1. Main block diagram of the proposed system

2.1. Dataset description

The dataset of this study has been collected from Kaggle, which has four classes: 'close', 'open', 'yawn', and 'no_yawn'. It has 2,175 files in these four categories. Table 1 shows the details of the drowsy driver detection dataset. This dataset shows that close and open categories consist of 617 images. On the other hand, yawn has 472 and no_yawn has 469 images. The sample images of the dataset are shown in Figure 2.

Table 1. Details of drowsy driver detection dataset

Class name	Quantity of images
Close	617
Open	617
Yawn	472
No_yawn	469
Total	2,175

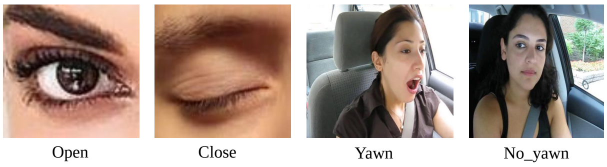

Figure 2. Sample images of the drowsy driver detection dataset

2.2. Pre-processing

Image pre-processing is one of the most crucial parts of computer vision tasks. In this study, we have used four different image pre-processing techniques: applying facial landmark predictor (FLP), data augmentation, data flipping, and data normalization.

2.2.1. Apply facial landmark predictor

FLP refers to facial landmark predictor. From the dataset description, it is seen that the yawn and not_yawn class has taken a major part of the human body including background. But to detect the yawn and no_yawn we just need the portion of mouth. So, we have used shape_predictor_68_face_landmarks.dat file to detect the portion of the mouth and cropped it. The block diagram of FLP is shown in Figure 3.

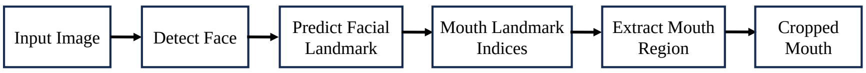

Figure 3. Block diagram of FLP

At first, FLP detects the face from in input image, then it predicts the facial landmark. After that, we have to define the mouth landmark indices. In this study, we have defined the mouth landmark indices as (48,48) and then extracted the mouth region. Finally, we saved it to our dataset. The result of before and after using FLP is shown in Figure 4.

Figure 4. Before and after using FLP

2.2.2. Data augmentation

Data augmentation is an essential part of computer vision tasks. It is used to increase the number of images and to bring the diversity of data. The dataset that we have used in this study consists of 2,175 images in four categories. Another issue was that there was a class imbalance problem in this dataset. Algorithm 1 illustrates the data augmentation technique for the proposed system.

Algorithm 1. Data augmentation algorithm

1. Define image folder path (fp)
2. Define the image number per folder (imgnum).
3. DataAugmentation (rot_rng, ws_rng, hs_rng, sh_rng, zm_rng)

for $i \leftarrow 1$ to fp **do**
  a. Directory creation (aug_fold_path).
  b. Load images
  c. Calculate the required number of aug_steps
  I. num_org_img = images of the main folder
  II. aug_steps = imgnum // num_org_img
  d. Initialize: $DataAugmentation$
  for $k \leftarrow 1$ to aug_steps **do**
    - Apply data augmentation using pre-defined parameters
    - Save augmented images to the aug_fold_path.
    - k++
  end for
end for
Output:
4. Print the Generated Images

In Algorithm 1, rot_rng, ws_rng, hs_rng, sh_rng, zm_rng stands for rotation range, width_shift_range, height_shift_range, shear_range and zoom_range respectively. num_org_img and aug_steps refer to the number of original image and augmentation steps. In this study, we have used an augmentation technique where we set,
rotation_range=10,
width_shift_range=0.1,
height_shift_range=0.1,
shear_range=0.1,
zoom_range=0.1,

In this algorithm, we have set the target number of images at first which is 2,000. After that, we calculate the number of original images in the folder. Then find out how many augmentation steps are needed to convert it to 2,000. After we used the data augmentation technique using the predefined parameters and saved the augmented images. Finally, we converted each category to 2,000 images and the total dataset images to 8,000.

2.2.3. Data flipping

Flipping is a common image processing technique used to create variations in training data. There are two types of flipping: horizontal and vertical. In this study, we have used horizontal_flip=True. An object on the left side of the original image will be on the right side of the horizontally flipped version.

2.2.4. Data normalization

Before supplying pictures to deep learning models or carrying out additional analysis, data normalization is frequently used as a preparatory step in image processing. Generally, the image pixels belong to 0 to 255 in an RGB image. In this study, we have converted and rescaled them into the range of [0,1].

2.3. Lightweight convolutional neural network for feature extraction

CNN has five main modules: convolutional layers, pooling layers, fully connected layers, activation layers, and backpropagation [22]. CNN has a major contribution to the field of computer vision and it is widely used in several domains such as segmentation, detection, and classification. In this study, we have proposed a lightweight CNN-based model to extract features from our image dataset. The block diagram of lightweight

CNN has been shown in Figure 5. The detailed parameter of the proposed lightweight CNN has been shown in Table 2.

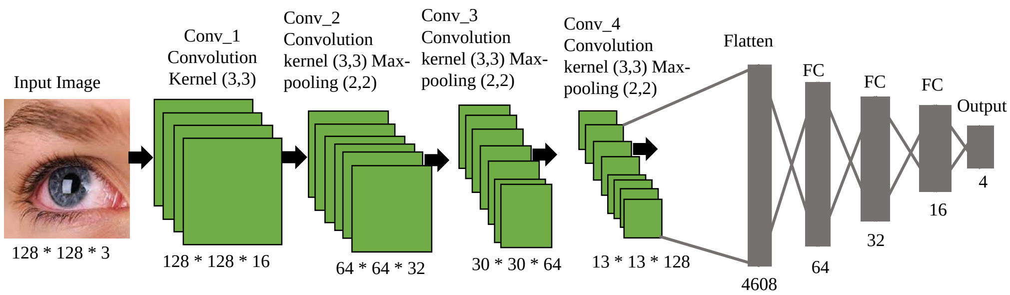

Figure 5. Block diagram of proposed lightweight CNN

Table 2. Detail parameter of proposed lightweight CNN model

Input shape	(128,128,3)				
Layer name	Filter size	Filters number	Pool_size	Output shape	Parameters
Conv2D	(3,3)	16	-	None, 128, 128, 16	448
max_pooling2d	-	-	(2, 2)	(None, 64, 64, 16)	0
conv2d_1	(3,3)	32	-	(None, 64, 64, 32)	4640
max_pooling2d_1	-	-	(2, 2)	(None, 32, 32, 32)	0
conv2d_2	(3,3)	64		(None, 30, 30, 64)	18496
max_pooling2d_2	-	-	(2, 2)	(None, 15, 15, 64)	0
conv2d_3	(3,3)	128	-	(None, 13, 13, 128)	73856
max_pooling2d_3	-	-	(2, 2)	(None, 6, 6, 128)	0
Flatten	-	-	-	(None, 4608)	0
dense	-	-	-	(None, 64)	294976
dense_1	-	-	-	(None, 32)	2080
Dense_2	-	-	-	(None, 16)	528
Dense_3	-	-	-	(None, 4)	68
Total parameters	395,092 (1.51 MB)				

In this proposed CNN model we have used four convolutional layers, four max-pooling layers, one flattened layer, and four fully connected layers. The input shape of the image is (128,128,3). The filter number of four convolutional layers are 16,32,64,128 respectively where the filter size is set to (3,3). In the max-pooling layer, we have set the pool size (2,2). In the fully connected layers, the number of units is 64,32,16,4 respectively. The total number of parameters of the proposed model is 3,95,092 which takes around 1.51 MB.

3. RESULT AND DISCUSSION

In this section, we have elaborately explained our model performance along with several curves and classification reports. The result using user input and comparison with other existing work is also shown here. Finally, we explained why we preferred our model as a lightweight model and how its performance varies with others.

3.1. Performance evaluation

The performance of the proposed system has been measured based on four modules: training and loss curve, confusion matrix, classification report, and receiver operating characteristic (ROC) curve [23].

3.1.1. Training and loss curve

The training and loss curve represents how model performances vary along with the epochs. In these curves, the y-axis represents the values of training or loss performance and the x-axis shows the number of epochs. The curve of model accuracy and model loss has been shown in Figure 6.

Figure 6. Model accuracy and loss curve

The model accuracy curve shows two lines one is yellow another is blue. The yellow line shows the validation accuracy whereas the blue line shows the performance metric of training accuracy. The total number of epochs is 10 which is shown in the x-axis. The values of training and validation accuracy at 1st epoch are 0.518 and 0.901. At the 10th epoch, the values of training and validation accuracy are 0.88 and 1.00.

Similarly, The model loss curve shows two lines one is yellow another is blue. The yellow line shows the validation loss whereas the blue line shows the values of training loss. The total number of epochs is 10 which is shown in the x-axis. The values of training and validation loss at 1st epoch are 0.789 and 0.3882. At the 10th epoch, the values of training and validation loss are 0.19 and 2.9299e-04 (closest to zero).

3.1.2. Classification report

A classification report, which is usually produced in Python using tools like scikit-learn, is a brief overview of several assessment criteria for a classification model [23]. It offers the most important performance indicators for every class in the categorization issue. Table 3 shows the classification report of the proposed system. This table shows the value of precision, recall, and f1-score for the four different classes open, close, yawn, and no-yawn. The values of precision, recall, and F1-score for the close class are 0.94, 0.96, and 0.95 respectively. Similarly, for the open class, these values are 0.98, 0.97 and 0.96; the No-yawn class consists of the values of 0.96, 0.98, and 0.97; and the last class Yawn has the values of 0.95, 0.92 and 0.93 respectively (see Table 3).

Table 3. Classification report of the proposed system

Class name	Precision	Recall	F1-score
Close	0.94	0.96	0.95
Open	0.98	0.97	0.96
No-yawn	0.96	0.98	0.97
Yawn	0.95	0.92	0.93

3.1.3. ROC curve

ROC stands for receiver operating characteristics curve plots the relationship between True positive and False positive rates [24], [25]. The true positive rate is shown on the y-axis whereas the false positive rate is shown on the x-axis. The ROC curve of the proposed system has been shown in Figure 7.

This figure shows area = 1.00. It means that it has returned a perfect classification. In other words, we can say that the proposed model perfectly classified the true positive and false positive instances.

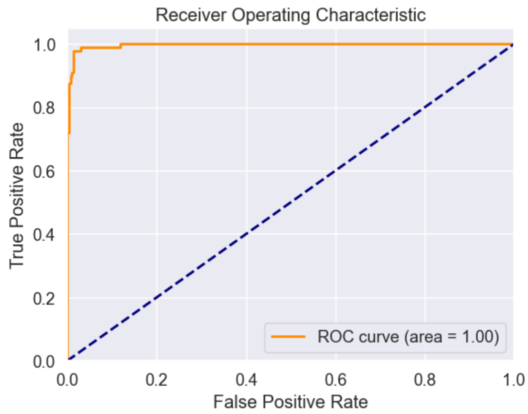

Figure 7. ROC curve of the proposed system

3.2. Result using random input

We have several user inputs from outside and inside of our dataset. The result using several user inputs is shown in Figure 8. This figure has four different input results indicated by Figures 8(a) to (d). In the case of Figure 8(a) the prediction value shows that the no-yawn class has higher values than others, so the prediction is no-yawn. Similarly for the input Figure 8(b) it predicts close, Figure 8(c) it returns open, and Figure 8(d) it returns yawn.

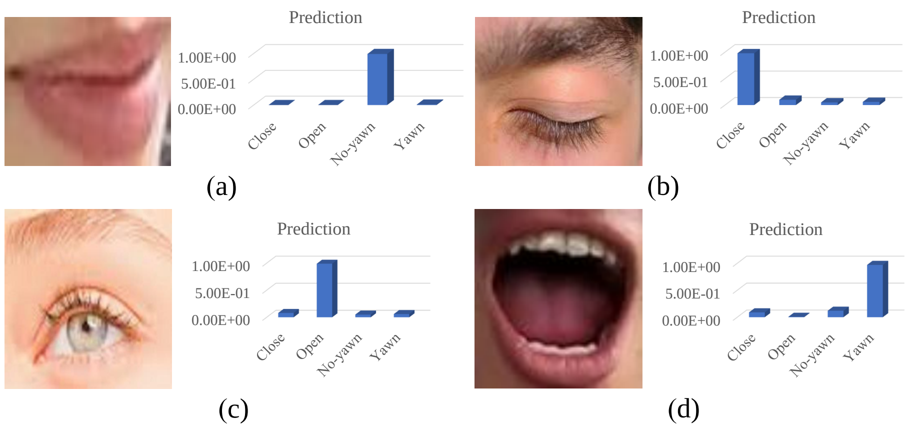

Figure 8. Prediction using several user inputs: (a) no-yawn, (b) close, (c) open, and (d) yawn

3.3. Comparison with existing methods

As drowsy driver detection is an important issue in reducing road accidents so many researchers already worked on this topic. In this section, we have compared our proposed system with some other previous work. Table 4 shows the comparison with other existing research.

Table 4. Comparison with other existing works

Article	Method	Features	Dataset	Accuracy (%)
Knapik and Cyganek [7]	Cold and hot voxels	Mouth	Own dataset	87
Kiashari *et al.* [8]	SVM and KNN	Respiration	Thermal image dataset	83
You *et al.* [11]	3D-deep CNN	Facial features	NTHUDDD dataset	73.9
Moujahid *et al.* [10]	Non-linear SVM	Eye, mouth, and head	NTHUDDD dataset	79.84
Our study	CNN	Yawn and eye	Kaggle dataset	96.6

Knapik and Cyganek [7] proposed a cold and hot voxels-based approach to detect drowsiness using their dataset. They have taken the features from the mouth and achieved an accuracy of 87%. Kiashari *et al.* [8]

introduced a drowsy driver detection system based on SVM and KNN. They have used a thermal image dataset and achieved an accuracy of 83%. You *et al.* [11] illustrated an approach based on 3d deep CNN where they used facial features to detect driver drowsiness. They used the NTHUDDD dataset and achieved an accuracy of 79.84%. Moujahid *et al.* [10] proposed a non-linear SVM to detect drowsy drivers. They used the NTHUDDD dataset and as features, they took eye, mouth, and head. The accuracy of their proposed model is 79.84%. Finally, the last row of the table shows the result of our proposed lightweight CNN where we have used the Kaggle dataset, and as the features we have taken yawn and eye. The accuracy of our proposed model is 96.6%.

3.4. Comparison with other model

The proposed lightweight CNN model has been compared with other existing models in terms of accuracy, number of parameters, and memory requirement where the proposed model has shown a satisfactory result. Table 5 shows the comparison with other models.

Table 5. Comparison with other model

No	Model name	Total parameters	Memory requirement (MB)	Accuracy (%)
1	VGG16	14,780,868	56.38	83
2	MobileNetV2	2,751,438	10.50	95
3	ResNet50	23,850,500	90.98	63
4	VGG19	20,090,564	76.64	77
5	DenseNet121	7,169,220	27.35	93.45
6	Our study	**395,092**	**1.51**	**96.6**

We have trained the dataset using several common existing models named VGG16, MobileNetV2, ResNet50, VGG19, and DenseNet121. Among this model, ResNet50 is the most-weighted model which has around 23 million parameters and a required memory size is 90.98 MB. Using ResNet50 we have achieved 63% accuracy. The second high-weighted model is VGG19 which has 20 million parameters and memory requirement is 76.64 MB. The accuracy using VGG19 is 77%. The third high-weighted model is VGG16, it has 14 million parameters and the accuracy is 83%. DenseNet121 and MobileNetV2 have fewer parameters compared to above mentioned three models. DenseNet121 has 7 million parameters and took a memory of 27 MB for training. The accuracy it shows is 93.45%. Among all pre-trained models MobileNetV2 performed very well, it has only 2 Million parameters and the required memory size is 10.50MB which shows an accuracy of 95%. Finally, the last one is our proposed lightweight CNN model that has around 4 lakhs (0.4 Million) parameters and memory requirement is 1.51 MB with a satisfactory accuracy of 96.6%.

4. CONCLUSION

In this paper, we have presented a CNN-based real-time drowsy driver detection system designed to enhance road safety by identifying drowsiness and microsleep episodes with high accuracy. Our proposed lightweight CNN architecture demonstrates superior performance, achieving an accuracy of 96.6% while maintaining efficiency with only 0.4 million parameters and a minimal memory requirement of 1.51 MB. This makes it particularly suitable for deployment in resource-constrained environments, such as Android applications. The results from our comparative analysis show that our model outperforms other popular CNN architectures, including VGG16, MobileNetV2, ResNet50, and DenseNet121, in both accuracy and resource efficiency. Future work includes expanding the dataset with diverse driver demographics and conditions for better generalizability and integrating sensor data like heart rate and steering patterns to enhance fatigue detection accuracy.

ACKNOWLEDGEMENTS

The manuscript's grammar was also improved with ChatGPT, which led to a notable improvement in its quality.

FUNDING INFORMATION

Authors state no funding involved.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Nippon Datta	✓	✓		✓		✓			✓	✓	✓			
Tanjim Mahmud	✓	✓		✓		✓			✓	✓	✓	✓		
Manoara Begum						✓				✓				
Mohammad Tarek Aziz						✓				✓				
Dilshad Islam						✓				✓				
Md. Faisal Bin Abdul Aziz						✓				✓				
Khudaybergen Kochkarov						✓				✓				
Temur Eshchanov						✓				✓				
Valisher Sapayev Odilbek Uglu						✓				✓				
Sobir Parmanov						✓				✓				
Mohammad Shahadat Hossain						✓				✓				
Karl Andersson					✓					✓				

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal Analysis
I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting
Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject Administration
Fu : **Fu**nding Acquisition

CONFLICT OF INTEREST STATEMENT

Authors state no conflict of interest.

DATA AVAILABILITY

This research utilizes publicly available data from the Yawn and Eye Dataset, accessible at the following link: https://www.kaggle.com/datasets/serenaraju/yawn-eye-dataset-new

REFERENCES

[1] T. Mahmud, B. Saha, D. Islam, M. T. Aziz, N. Datta, K. Barua, M. S. Hossain, and K. Andersson, "Deep learning approach for driver drowsiness detection in real time," in *Innovations in Cybersecurity and Data Science*, Singapore: Springer Nature Singapore, 2024, pp. 775–790, doi: 10.1007/978-981-97-5791-6_56.

[2] A. Pinto, M. Bhasi, D. Bhalekar, P. Hegde, and S. G. Koolagudi, "A deep learning approach to detect drowsy drivers in real time," in *2019 IEEE 16th India Council International Conference (INDICON)*, Rajkot, India, 2019, pp. 1-4, doi: 10.1109/INDICON47234.2019.9030305.

[3] T. Vesselenyi, S. Moca, A. Rus, T. Mitran, and B. Tătaru, "Driver drowsiness detection using ann image processing," in *IOP Conference Series: Materials Science and Engineering*, vol. 252, no. 1, 2017, p. 012097, doi: 10.1088/1757-899X/252/1/012097.

[4] A. Amodio, M. Ermidoro, D. Maggi, S. Formentin, and S. M. Savaresi, "Automatic detection of driver impairment based on pupillary light reflex," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 8, pp. 3038-3048, Aug. 2019, doi: 10.1109/TITS.2018.2871262.

[5] J. H. Yang, Z.-H. Mao, L. Tijerina, T. Pilutti, J. F. Coughlin, and E. Feron, "Detection of driver fatigue caused by sleep deprivation," *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*,vol. 39, no. 4, pp. 694-705, July 2009, doi: 10.1109/TSMCA.2009.2018634.

[6] S. Hu and G. Zheng, "Driver drowsiness detection with eyelid related parameters by support vector machine," *Expert Systems with Applications*, vol. 36, no. 4, pp. 7651–7658, 2009, doi: 10.1016/j.eswa.2008.09.030.

[7] M. Knapik and B. Cyganek, "Driver's fatigue recognition based on yawn detection in thermal images," *Neurocomputing*, vol. 338, pp. 274–292, 2019, doi: 10.1016/j.neucom.2019.02.014.

[8] S. E. H. Kiashari, A. Nahvi, H. Bakhoda, A. Homayounfard, and M. Tashakori, "Evaluation of driver drowsiness using respiration analysis by thermal imaging on a driving simulator," *Multimedia Tools and Applications*, vol. 79, pp. 17 793–17 815, 2020, doi: 10.1007/s11042-020-08696-x.

[9] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, San Diego, CA, USA, 2005, pp. 886-893 vol. 1, doi: 10.1109/CVPR.2005.177.

[10] A. Moujahid, F. Dornaika, I. Arganda-Carreras, and J. Reta, “Efficient and compact face descriptor for driver drowsiness detection,” *Expert Systems with Applications*, vol. 168, p. 114334, 2021, doi: 10.1016/j.eswa.2020.114334.

[11] F. You, Y. Gong, H. Tu, J. Liang, and H. Wang, “A fatigue driving detection algorithm based on facial motion information entropy,” *Journal of Advanced Transportation*, vol. 2020, pp. 1–17, 2020, doi: 10.1155/2020/8851485.

[12] A. S. Zandi, A. Quddus, L. Prest, and F. J. Comeau, “Non-intrusive detection of drowsy driving based on eye tracking data,” *Transportation Research Record*, vol. 2673, no. 6, pp. 247–257, 2019, doi: 10.1177/0361198119847985.

[13] Z. Mardi, S. N. M. Ashtiani, and M. Mikaili, “Eeg-based drowsiness detection for safe driving using chaotic features and statistical tests,” *Journal of Medical Signals and Sensors*, vol. 1, no. 2, pp. 130-137, 2011.

[14] S. M. R. Noori and M. Mikaeili, “Driving drowsiness detection using fusion of electroencephalography, electrooculography, and driving quality signals,” *Journal of Medical Signals and Sensors*, vol. 6, no. 1, p. 39, 2016.

[15] J. Krajewski, D. Sommer, U. Trutschel, D. Edwards, and M. Golz, “Steering wheel behavior based estimation of fatigue,” in *Driving Assessment Conference*, 2009, vol. 5, no. 2009, pp. 118-124, doi: 10.17077/drivingassessment.1311.

[16] T. Danisman, I. M. Bilasco, C. Djeraba, and N. Ihaddadene, “Drowsy driver detection system using eye blink patterns,” in *2010 International Conference on Machine and Web Intelligence*, Algiers, Algeria, 2010, pp. 230-233, doi: 10.1109/ICMWI.2010.5648121.

[17] S. Abtahi, B. Hariri, and S. Shirmohammadi, “Driver drowsiness monitoring based on yawning detection,” in *2011 IEEE International Instrumentation and Measurement Technology Conference*, Hangzhou, China, 2011, pp. 1-4, doi: 10.1109/IMTC.2011.5944101.

[18] K. Dwivedi, K. Biswaranjan, and A. Sethi, “Drowsy driver detection using representation learning,” in *2014 IEEE International Advance Computing Conference (IACC)*, Gurgaon, India, 2014, pp. 995-999, doi: 10.1109/IAdCC.2014.6779459.

[19] S. Damian, J. Dybała, P. Czapski, and P. Skalski, “Driver drowsiness detection systems,” *Proceedings of the Institute of Vehicles*, 2017, vol. 112, no. 3, pp. 41-48.

[20] P. Dey, T. Mahmud, M. S. Chowdhury, M. S. Hossain, and K. Andersson, “Human age and gender prediction from facial images using deep learning methods,” *Procedia Computer Science*, vol. 238, pp. 314–321, 2024, doi: 10.1016/j.procs.2024.06.030.

[21] N. S. Nuha, T. Mahmud, N. Rezaoana, M. S. Hossain, and K. Andersson, “An approach of analyzing classroom student engagement in multimodal environment by using deep learning,” in *2023 IEEE 9th International Women in Engineering (WIE) Conference on Electrical and Computer Engineering (WIECON-ECE)*, Thiruvananthapuram, India, 2023, pp. 286-291, doi: 10.1109/WIECON-ECE60392.2023.10456488.

[22] S. U. Habiba *et al.*, “Early prediction of chronic kidney disease using machine learning algorithms with feature selection techniques,” in *International Conference on Applied Intelligence and Informatics*, Springer, 2023, pp. 224–242, doi: 10.1007/978-3-031-68639-9_14.

[23] A. D. Bappy, T. Mahmud, M. S. Kaiser, M. Shahadat Hossain, and K. Andersson, “A bert-based chatbot to support cancer treatment follow-up,” in *International Conference on Applied Intelligence and Informatics*, Springer, 2023, pp. 47–64, doi: 10.1007/978-3-031-68639-9_4.

[24] S. J. Schmugge, S. Jayaram, M. C. Shin, and L. V. Tsap, “Objective evaluation of approaches of skin detection using ROC analysis,” *Computer Vision and Image Understanding*, vol. 108, no. 1–2, pp. 41–51, Oct. 2007, doi: 10.1016/j.cviu.2006.10.009.

[25] R. V. Anand, R. VijayaKumar, and M. Rajesh, “Advanced Traffic monitoring:Precise vehicle counting,speed estimation,color recognition and Type classification utilizing ROC curve analysis,” *2024 10th International Conference on Communication and Signal Processing (ICCSP)*, pp. 349–354, Apr. 12, 2024. doi: 10.1109/iccsp60870.2024.10543401.

BIOGRAPHIES OF AUTHORS

Nippon Datta received a B.Sc. Engineering Degree from the Rangamati Science and Technology University, Rangamati-4500, Bangladesh. Currently, he is an M.Sc Engineering student at Chittagong University of Engineering and Technology, Chattogram, Bangladesh, under the Department of Computer Science and Engineering. His research interests include digital image processing and AI in healthcare. He can be contacted at email: nipponrmstu.cse@gmail.com.

Tanjim Mahmud a Member of IEEE, received his Ph.D. in Engineering with a specialization in Natural Language Processing from the Kitami Institute of Technology, Japan. He also earned his M.Sc. in Engineering and B.Sc. in Computer Science and Engineering from the University of Chittagong, Bangladesh. He serves as an assistant professor in the Department of Computer Science and Engineering at Rangamati Science and Technology University, Bangladesh. His research areas encompass artificial intelligence, natural language processing, AI applications in healthcare and agriculture, machine learning, and image processing. He can be contacted at email: tanjim_cse@yahoo.com.

Manoara Begum earned her M.Sc. and B.Sc. degrees in Computer Science and Engineering from the University of Chittagong, Bangladesh. She currently serves as an Assistant Professor in the Department of Computer Science and Engineering at Port City International University, Bangladesh. Her research focuses on machine learning, deep learning, and natural language processing. She can be contacted at email: manoara.cse34@gmail.com.

Mohammad Tarek Aziz received a B.Sc. Engineering Degree from the Rangamati Science and Technology University, Rangamati-4500, Bangladesh. Currently, he is an M.Sc. Engineering student at Chittagong University of Engineering and Technology, Chattogram, Bangladesh, under the Department of Computer Science and Engineering. His research interests include machine learning, deep learning, digital image processing, and AI in healthcare. He can be contacted at email: tarekaziz4288@gmail.com.

Dilshad Islam holds both an M.Sc. and B.Sc. in Computer Science and Engineering from the University of Chittagong, Bangladesh. She currently serves as an Associate Professor in the Department of Physical and Mathematical Sciences, Chattogram Veterinary and Animal Sciences University, Bangladesh. Her research areas encompass artificial intelligence and image processing. She can be contacted at email: 6dilshad@gmail.com.

Md. Faisal Bin Abdul Aziz holds both an M.Sc. and B.Sc. in Computer Science and Engineering from the University of Chittagong, Bangladesh. He serves as an assistant professor in the Department of Computer Science and Engineering at Comilla University, Bangladesh. His research areas encompass artificial intelligence and image processing. He can be contacted at email: faisal@cou.ac.bd.

Khudaybergen Kochkarov research field is technical subjects from Tashkent State University of Economics, Tashkent, Uzbekistan. He is currently working in the Department of Special Science at Tashkent State University of Economy, Tashkent, Uzbekistan. He has a postdoctoral fellowship from Urgench State University, Urgench, Uzbekistan. His research interests include telecommunications, image processing, computer vision, and deep learning. He can be contacted at email: xquchqarov@tsue.uz

Temur Eshchanov research field is technical subjects from Urgench State University, Urgench, Uzbekistan. He is currently working in the Department of Special Science at the Tashkent State University of Economy, Tashkent, Uzbekistan. He has a postdoctoral fellowship from Urgench State University, Urgench, Uzbekistan. His research interests include telecommunication, image processing, computer vision, and deep learning. He can be contacted at email: temur@urdu.uz.

Valisher Sapayev Odilbek Uglu received his Ph.D. degree in teaching technical subjects from the National University of Uzbekistan, Tashkent, Uzbekistan. He is currently the head of the Career Department at Mamun University, Khiva, Uzbekistan. He has a postdoctoral fellowship from Urgench State University, Urgench, Uzbekistan. His research interests include teaching telecommunication, image processing, computer vision, and deep learning. He can be contacted at email: sapayev.vali.2017@gmail.com.

Sobir Parmanov studied at the National University of Uzbekistan, Tashkent, Uzbekistan. He is currently working in the Department of Technology at the National University of Uzbekistan, Tashkent, Uzbekistan. He is a Postdoctoral fellowship with the National University of Uzbekistan, Urgench, Uzbekistan. His research interests include telecommunication, image processing, computer vision, and deep learning. He can be contacted at email: parmonov_s@nuu.uz.

Mohammad Shahadat Hossain a Senior Member of IEEE, earned his M.Phil. and Ph.D. degrees in computation from the Institute of Science and Technology at the University of Manchester (UMIST), U.K., in 1999 and 2002, respectively. He is currently a Professor of Computer Science and Engineering at the University of Chittagong, Bangladesh, and serves as a Visiting Professor at Luleå University of Technology, Sweden. His research interests focus on e-government, risk and uncertainty modeling through evolutionary computing, the development of pragmatic software tools and methods, information systems, and expert systems. He can be contacted at email: hossain_ms@cu.ac.bd.

Karl Andersson a Senior Member of IEEE, obtained his M.Sc. in Computer Science and Technology from the Royal Institute of Technology in Stockholm, Sweden, and his Ph.D. in Mobile Systems from Luleå University of Technology, Sweden. Following postdoctoral work at the Internet Real-Time Laboratory, Columbia University, New York, USA, and the National Institute of Information and Communications Technology in Tokyo, Japan, he now serves as a Professor at the Cybersecurity Laboratory at Luleå University of Technology. His research focuses on green and mobile computing, the internet of things, cloud technologies, and information security. He can be contacted at email: karl.andersson@ltu.se.