IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 12 July 2024, accepted 29 July 2024, date of publication 8 August 2024, date of current version 19 August 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3440585

RESEARCH ARTICLE

DrowsyDetectNet: Driver Drowsiness Detection Using Lightweight CNN With Limited Training Data

MADDURI VENKATESWARLU AND VENKATA RAMI REDDY CH
School of Computer Science and Engineering, VIT-AP University, Amaravati 522237, India
Corresponding author: Venkata Rami Reddy Ch (venkataramireddy.chirra@vitap.ac.in)

This work was supported by VIT AP University, Amaravati, Andhra Pradesh.

ABSTRACT One major factor in the rising incidence of traffic accidents is the driver's drowsiness. Innovations in computer vision technology have made it possible to construct smart cams that can recognize driver fatigue. By alerting drivers, this technology successfully lowers the total number of accidents caused by weariness. This study proposes a DrowsyDetectNet that utilizes a shallow Convolutional Neural Network (CNN) architecture to identify driver drowsiness. The 68-point face landmark identification approach is used to identify faces and extract eye areas. The proposed system employs a shallow CNN architecture with fewer layers and parameters to detect driver drowsiness with limited training data. Feature extraction focuses on relevant visual cues for drowsiness detection, such as eyelid closure. The transfer learning models, such as VGG19, ResNet50, MobileNetV2, and InceptionV3, are also used to identify driver drowsiness. Two datasets, Dataset-1 and Dataset-2, were utilized to assess this study. On two datasets, the proposed DrowsyDetectNet produced an accuracy of 99.23% and 99.14%, respectively. The proposed DrowsyDetctNet framework achieved better accuracy when compared with state-of-the-art models and pre-trained models.

INDEX TERMS Drowsiness detection, DrowsyDetectNet, shallow CNN, limited training data, pre-trained models.

I. INTRODUCTION

Drowsiness is an unpleasant feeling of being excessively weary or drowsy during the day, which might make you forget things or make you nod off unnecessarily. This is a common event that may cause people to become distracted and put motorists in danger. Any of two methods can be used to determine whether someone is sleepy or not. The first includes non-intrusive approaches, whereas the second includes intrusive approaches, such as patterns of lateral acceleration, lateral displacement, steering wheel motion, and break patterns, without any need for direct physical contact. These patterns can be changed based on the road conditions [1]. Intrusive approaches, like electroencephalography (EEG) and electrocardiography (ECG), which assesses the driver's physiological factors, including pulse rate and heartbeat, can be used to ascertain whether a driver is drowsy or not [2]. Physiological signals can accurately and promptly indicate a driver's level of exhaustion, but collecting these signals necessitates the use of numerous sensors that come into touch with the driver [3], [4]. Intrusive techniques provide more precise detection than non-intrusive ones. However, it might be challenging to apply these strategies in practical situations [5].

Numerous car accidents, injuries, and fatalities have been associated with drowsiness. This highlights how important it is to have systems that monitor driver drowsiness and issue alerts when it occurs—according to approximations furnished by the National Highway Traffic Safety Administration (NHTSA), driving while fatigued caused $12.5 billion in economic loss, 71,000 injuries, and 1,550 fatalities. Statistics (NHTSA) [6] show that in 2017, accidents involving drowsy drivers resulted in 50,000 injuries and 795 fatalities. Researchers believe that it is crucial to be able to recognize signs of fatigue based on behavioral indicators, such as changes to the lips, eyes, or other facial characteristics.

The associate editor coordinating the review of this manuscript and approving it for publication was Abdullah Iliyasu.

By analyzing these indicators, researchers want to create tools for spotting driver weariness and putting safety precautions in place to avoid accidents [7], systems for identifying driver inebriation [8], human-computer interaction (HCI) [9], facial expression recognition (FER) [10], brain-computer interface (BCI) [11], healthcare [12], etc. can be designed and developed more easily with the use of eye state detection systems. Most of the applications make use of eye status data, both directly and indirectly.

Several computer vision-based methods for sleepiness detection have been developed over the past few decades to monitor driver alertness. Eyes closing, nodding, and yawning are all examples of facial expressions that can indicate sleepiness. Drivers' eye closure frequency and duration increase and their eye open frequency and duration decrease when they are fatigued [13].

Researchers are making significant advancements in drowsiness detection technology. New deep-learning techniques addressed pose variations and incorporated mouth and eye features for improved accuracy [21], [34]. Lightweight models and hierarchical frameworks were also being developed for real-time applications and specific environments like suburban roads [30], [31]. To ensure real-world effectiveness, research focuses on evaluating robustness against challenges like occlusions and generalizability across various conditions and populations [27], [29]. Furthermore, the field is expanding beyond car drivers by developing drowsiness detection models for crane operators, highlighting its potential for diverse applications [22]. While our study introduces the DrowsyDetectNet framework, there's still a gap in literature regarding the development of lightweight models for this purpose. Current methods often relied on complex architectures unsuitable for resource-constrained environments like vehicle systems. Hence, there's a crucial need for lightweight models with reduced computational complexity, that maintain high accuracy in drowsiness detection. Through exploring novel approaches, such as our proposed shallow CNN architecture with fewer layers, and utilized limited dataset, can address this gap and enhance driver safety systems.

A. MOTIVATION

The following are a few major reasons for creating and using driver fatigue detection technology:

- **Safety:** Monitoring in real-time driver fatigue can assist in averting collisions by informing drivers of their impairment and enabling them to adopt the appropriate safety measures.
- **Reducing Accidents:** Accidents can be significantly decreased by identifying tiredness early.
- **Improved Productivity:** Systems that detect drowsiness may preserve users' concentration, lower accidents, and boost overall productivity.
- **Cost Savings:** Drowsiness detection devices can save money for both individuals and businesses by reducing accidents.
- **Technological Advancements:** The need for drowsiness detection systems in vehicles has surged as a result of recent developments in sensor technology, machine learning, and artificial intelligence, which have made it feasible to build accurate and fairly cost-effective systems.

B. CONTRIBUTIONS

- Designed a shallow CNN architecture with fewer layers to determine driver fatigue depending on eye state.
- Identified the face in a driver image utilizing the 68-point face landmark detector.
- Utilized the 68-point face landmark detector to identify eye area after the face has been detected.
- The effects of hyperparameters such as batch size, number of epochs, optimizers, and learning rate were tuned on the proposed shallow CNN model.
- The performance of the suggested work was evaluated using a variety of metrics, including the Receiver Operator Characteristic (ROC) curve, Precision-Recall (PR) curve, accuracy, precision, F1-Score, recall, confusion matrix, and so on.

One of the main contributions of this paper is the novelty of the shallow CNN architecture for driver drowsiness detection. Unlike models like VGG19, InceptionV3, MobileNetV2, and ResNet50, the shallow CNN is lightweight, computationally efficient, and optimized for limited training data, focusing on visual cues like eyelid closure. Its efficiency makes the ideal for real-time applications on embedded systems.

The following is the manuscript's structure: The second section discusses the sleepiness detection literature; the third section discusses the components and methods used in the suggested system; and the fourth section presents the results of the experiments along with a breakdown of each CNN design. The fifth section concludes with the recommendations and findings for further study.

II. RELATED WORK

During this study, Phan, et al. [14] for drowsiness detection involved testing and training phases. In the training stage, footage captured by a vehicle's security system is preprocessed to detect faces and head regions using a specific network. These extracted images are then used to train deep neural networks, like Inception-V3, DenseNet, LSTM, and VGG-16, with improvements made to their layers for drowsiness detection. In the testing phase, the trained models are evaluated on a separate dataset to identify sleepy conditions with a 98% accuracy rate. Faisal et al. [15] proposed a real-time CNN-based system for detecting driver drowsiness. The system starts by capturing frames of the face of the driver using a camera and, after that, detects the location of the eyes. The technology detects whether the driver is tired or not and notifies them depending on predetermined parameters. The CNN model is identified through image extraction, preprocessing, and the optimization of hyperparameters such as order of kernel, learning speed, maximum pool size, and epochs. The CNN's trained model is assessed on the dataset utilizing a training efficiency of 99.87% and a testing accuracy of 97.98%. Ganguly et al. [16] proposed detection systems that make use of a traditional CNN and a faster region-based CNN. These are the two deep learning frameworks. The system first detects eye regions using the Faster region-based CNN, which consists of convolutional neural networks and max-pooling layers. Then, the Fast RCNN detector utilizes proposed areas to generate object proposals and estimate the probability of object detection. Finally, the eye states are detected and classified using layers of pooling and convolution in the classical CNN.

Magan et al. [17] proposed using image sequences that use the driver's facial features to build a system that gauges their level of fatigue. The device, which is a component of a driver-based ADAS system, aims to minimize false positives while maximizing early fatigue detection. The system uses 10 frames per second (FPS) to capture 600 frames over a 60-second period, which are then processed and analyzed to assess the level of drowsiness and activate appropriate alarms if necessary. Florez et al. [18] proposed six steps in the process of identifying driver drowsiness: data acquisition, pre-processing of video frames using facial landmark detection, constructing a dataset, testing trained models, training CNN architectures, and forecasting driver fatigue. The pre-processing step includes a methodology for selecting a region of interest (ROI) surrounding the eyes by calculating distances between facial points, ensuring the ROI captures relevant information even during head movements. The trained models are evaluated, and the best-performing model is used for driver drowsiness prediction. Jahan et al. [19] proposed using a customized CNN model named 4D to identify sleepiness depending on the eye condition. The model consists of various layers: convolution, activation, batch normalization, dropout, max-pooling, fully connected, and output. Additionally, this paper mentions the use of transfer learning CNN models, specifically VGG19 and VGG16, for image classification tasks. The MRL Eye dataset, comprising 47,173 images of both open and closed eyes, was utilized in this instance to train the model, resulting in an accuracy of 97.53%. Akrout and Fakhfakh [20] suggested a system that tracks eye area, estimates facial landmarks, and calculates head posture using the Media Pipe Face Mesh. Subsequently, they utilize an innovative method for iris identification and normalization, succeeded by MobileNetV3 architecture-based feature extraction. The resulting features, including distances between various facial points and head angles, are fed into a deep LSTM network to detect driver fatigue. Additionally, it discusses the analysis of the iris and its surroundings, including the segmentation and normalization of the iris for feature extraction.

Kumar et al. [21] proposed an approach that utilizes a hybrid deep learning approach, combining InceptionV3 and LSTM, to analyze the mouth and eye regions for spatial feature extraction. The modified InceptionV3 incorporates a global average-pooling layer and dropout layer to enhance adaptability and prevent over-fitting, respectively. The outcome of the InceptionV3 modification is then fed into LSTM for determining if the driver is drowsy or not, with an accuracy of 93.69%. Liu et al. [22] proposed a workflow and hybrid deep neural network design for fatigue detection in crane operators. The workflow involves capturing videos, detecting operators'faces, extracting facial landmarks, and extracting fatigue features for training fatigue classifiers. The three primary modules of the architecture—he Face Detector, the Extractor of Spatial Features using MobileNet, and the Time-Based Characteristic Modeling using LSTM—are coupled by learning networks to determine the degree of fatigue and, if necessary, initiate alerts. Mu et al. [23] proposed a technique used to eliminate interference factors such as noise and uneven lighting in collected images. Common image noises include Gaussian noise and impulse noise, and techniques for filtering like Gaussian, median, and mean filters are employed to reduce their influence. Additionally, human eye state recognition methods, such as the Hough transform, which is useful for determining the state of the human eye based on detecting the presence or absence of a circle, indicating an open or closed eye respectively.

Phan et al. [24], two techniques for drowsy alert systems, were proposed. The first approach does away with the requirement for pre-determination by using facial landmarks, blink, and yawn features (EAR, LIP) are computed to determine customizable thresholds for every driver. The second approach builds adaptable deep neural networks with changes in certain layers to improve drowsiness detection by utilizing cutting-edge deep learning techniques, such as SSD-ResNet-10, derived from MobileNet-V2 and ResNet-50V2. Transfer learning is applied to achieve faster learning, improve classification accuracy, and eliminate the requirement for large training datasets. Zhu et al. [25] suggested a methodology for driver fatigue recognition utilizing the algorithms of the tasks-constrained deep convolutional network (TCDCN). Drivers using sunglasses or glasses-wearing facial feature images are used to train the algorithm offline, and it performs multiple tasks simultaneously, such as gender identification, face position detection, and glasses recognition. The TCDCN algorithm shows potential in handling challenges like occlusion and position change, and it utilizes parameter sharing across multiple subnetworks to make use of all available information for accurate face detection. Abbas et al. [26] recommended the deep learning architecture ReSVM to detect driver attention. A support vector machine (SVM) is fed deep features for categorization using ReSVM, an enhanced ResNet-50 version. The deep features are taken through the ResNet-50 pooling layer. The ReSVM model takes images of different sizes and lighting conditions as input, stacks the features obtained from the final ResNet-50 convolution layer, and determines the feature map's mean before classifying the data using SVM.

Jia et al. [27] developed a system based on a deep learning approach for detecting driving weariness. The improved Convolutional Neural Network(MTCNN) for multitasking is utilized by finding the driver's face to detect facial cues of significance. The algorithm also incorporates techniques such as adding a Spatial Pyramid Polling (SPP) layer to the network structure and the Batch Normalization(BN) algorithm is applied to improve network accuracy and performance. Mohamed et al. [28] explained the role of deep learning algorithms and the data sources and data augmentation techniques used in the study. It also briefly describes the technical summary of the deep learning algorithms considered and the assessment metrics that are employed for measuring their performance. The configuration of the study, including the datasets used and their properties, as well as the training and testing process, is also discussed. Dua et al. [29] proposed a driver fatigue identification system consisting of four models: FlowImageNet, AlexNet, VGGFaceNet, and ResNet. To feed these models, frames are taken out of the input video stream. Each model is trained to learn specific features related to drowsiness, such as behavioral, environmental, facial, and hand gesture features. The outcomes of these models are then combined using an ensemble strategy to categorize the video as either sleepy or awake.

Jamshidi et al. [30] proposed an approach for drowsiness identification in drivers that utilizes a driver-based approach, obtaining visual information from the vehicle's camera. The framework consists of four main phases: facial recognition, detecting the condition of the eyes and mouth, situation recognition, and tiredness identification. The technique uses a hierarchical framework to identify pertinent data, like the presence of glasses and illumination. It utilizes a network for situation detection to increase the accuracy of mouth and eye state detection. Saurav et al. [31] proposed an approach for eye state recognition that entails multiple steps, like eye patch extraction, pre-training 2 CNN models (CNN Model1 and CNN Model2) on a facial emotion dataset, individual and ensemble fine-tuning of the transfer learning models to eye state datasets, and intended eye state datasets. The DCNNE model integrates the learned features from both CNN models to create a stronger eye state classifier. Bajaj et al. [32] proposed various approaches to develop an effective model for identifying driver drowsiness. In this study, they mention the publication trends in driver drowsiness detection systems, with a focus on the interest in developing countries like India. Additionally, it provides information on the hardware components used, including the Raspberry Pi 3 model B+ and other sensors, for implementing the hybrid model.

Flores-Monroy et al. [33] suggested a real-time technique for identifying driver fatigue consists of several stages, including face identification employing the Viola & Jones formula, face analysis employing a shallow CNN (SS-CNN) that has been specially created, and consecutive results analysis. The SS-CNN is designed to categorize the face region into open and closed eyes. The selected configuration of the SS-CNN has roughly 600K trainable parameters, enabling real-time operation utilizing a compact GPU system. Chirra et al. [34] suggested a deep CNN-based technique for identifying drowsiness that extracts eye regions and detects faces using the Viola-Jones face detection method. After feeding these eye areas into a CNN with four convolutional layers for feature extraction, the images are classified as drowsy or not using a Softmax layer. Using test data samples with an accuracy of 96.42%, the suggested approach proved to be successful in identifying driver fatigue depending on eye state.

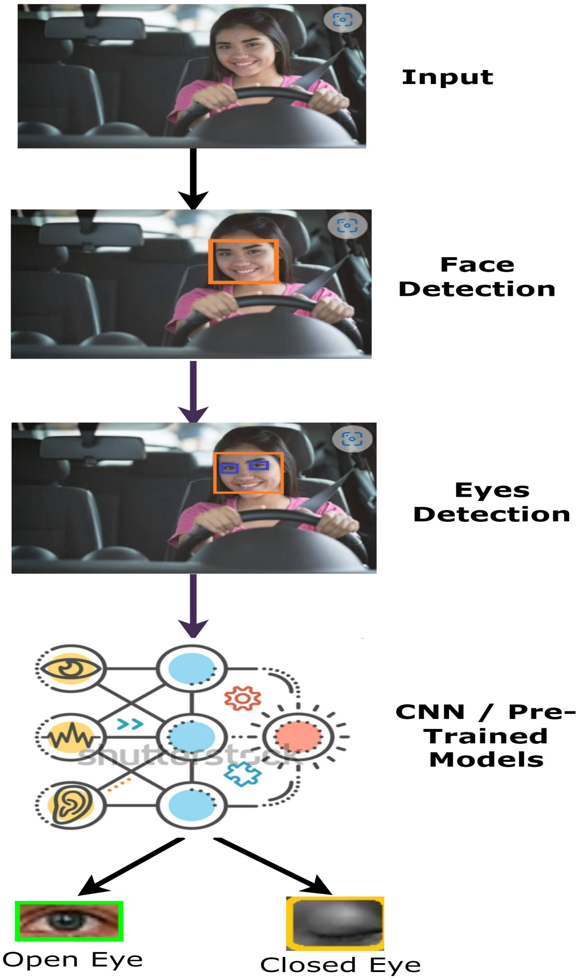


FIGURE 1. Proposed DrowsyDetectNet framework.

III. METHODOLOGY

A. PROPOSED DrowsyDetectNet FRAMEWORK

This work aims to develop a DrowsyDetectNet framework to find out whether a driver is drowsy or not. FIGURE 1 renders the suggested system architecture. To determine the driver's face location for an input image or video, the 68-point facial landmark detection algorithm is employed. Subsequently, the eye region is removed from the face. To identify an "open eye" or "closed eye, " that extracted eye image is loaded into a suggested shallow CNN model.

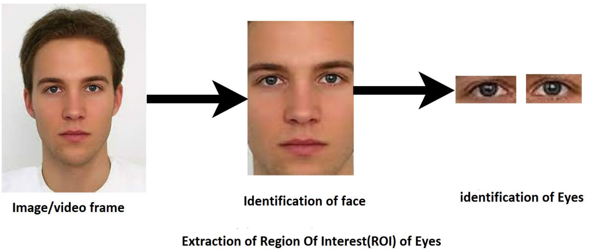


FIGURE 2. Extraction of region of interest (ROI).

B. RECOGNITION OF FACES AND EYE REGION EXTRACTION

The Region of Interest(ROI) extraction process involves isolating and analyzing specific areas within an image that contains relevant information about the eyes, which is depicted in FIGURE 2. To identify the drowsiness of a driver, no need to use the full face, but merely the eye region. The 68 (x, y) positions corresponding to the face's facial structures are estimated using the Dlib library's facial landmark detector. In FIGURE 2, the following represents the 68 coordinates: The jaw ranges from 1 to 17, the right and left eyebrows range from 18 to 22, and 23 to 27, the nose ranges from 28 to 36, the right and left eyes range from 37 to 42 and 43 to 48, the mouth ranges from 49 to 60, and the lips from 61 to 68.

The process involved detecting and cropping eyes from an image using the Dlib library for face detection and landmark prediction, along with OpenCV for image manipulation.

First, the shape predictor extracts the 68 facial landmarks for the current face. Specific indexes within these 68 points correspond to the left and right eye regions. The x and y coordinates of each eye corner are extracted based on the indexes 43 to 48 of the left eye and 37 to 42 of the right eye. Using these coordinates, the left and right eye regions are cropped from the image. These cropped eye regions are then provided to the model to identify drowsiness.

The *Figure 3* [39] can be downloaded at The-face-shape-with-68-landmarks.

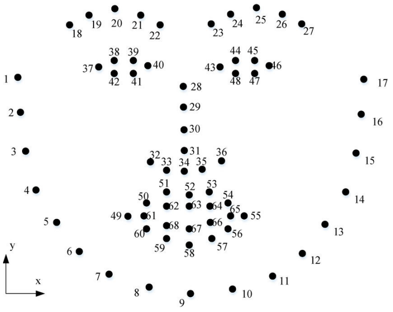


FIGURE 3. The-face-shape-with-68-landmarks.

This study uses direct eye images, allowing the model to extract features like eyelid closure directly to detect drowsiness.

The process detailed in Algorithm 1, from detecting facial landmarks to identifying the eye regions, is primarily by the 68-point face landmarks detector algorithm. Users supply a picture of a human face, which the pre-trained detector uses to identify 68 distinct landmarks that define the main features of the face, such as the mouth, nose, eyes, and contours of the face. The algorithm determines the indices corresponding to the eyes of the left (indices 37 to 42) and right (indices 43 to 48), calculating the bounding box coordinates to define the eye regions. While the identification steps are, cropping of the eye regions from the image is performed manually using the calculated coordinates. These manually cropped eye regions are subsequently fed into a shallow CNN model, which processes the eye regions to detect signs of drowsiness, particularly focusing on eyelid closure. This approach combines detection for accuracy with manual intervention for precise preparation of input data for the drowsiness detection model.

C. DESIGN OF PROPOSED SHALLOW CNN MODEL

The convolution process begins in the upper left corner of the supplied image, scanning horizontally until it covers the entire row, then moves downwards to repeat the process. The output values of this operation would create the feature map, which is specified by equation(1):

$$\begin{aligned} X(m,n) &= (I \times K)[m,n] \\ &= \sum_{a} \sum_{b} I[a,b] \times K[i-a, j-b] \end{aligned} \tag{1}$$

where K stands for the kernel, I for the input image, X for the feature map, m for the index rows of the convolved matrix, and n for the index columns.

The input image's size I(m × n) determines the size of the feature-map X, the filter K(u × v), and several strides g within the image provided by equation(2).

$$Y(i,j) = Y\left(\frac{m-u}{g} + 1, \frac{n-v}{g} + 1\right) \tag{2}$$

The CNN architecture utilized in the suggested system to determine the factors that contribute to drowsiness is schematically represented in FIGURE 4. The proposed shallow CNN comprises four convolutional blocks, four max-pooling layers, two dropout layers, and two FC layers.

To create thirty-two 128 × 128 feature maps, thirty-two 3 × 3 filters are convolved with an input image of 128 × 128 pixels. Following 2 × 2 max-pooling processes, reduces the spatial dimensions by taking the maximum value in each window, retaining important spatial features and reducing computational complexity. These feature maps are reduced to thirty-two 64 × 64 feature maps. Next, downsized feature maps are passed to the dropout layer with 0.2. Sixty-four 64 × 64 feature maps are produced in the 2nd convolution layer by utilizing 64 filters. The generated feature maps are reduced to sixty-four 32 × 32 feature maps by the second 2 × 2 max-pooling layer. The downsized feature maps are then fed to the dropout layer with 0.2. Next, the outcome of the dropout

Algorithm 1 Extracting the Region of Interest (ROI)

```
input:
• Image, I contain a human face.
• 68-point face landmarks vector L.
Output:
• ROI_Left_Eye, ROI_Right_Eye
Facial Landmark Detection:
  • Use a robust facial landmark detector to obtain the 68-point landmark vector L from an image I:
    – Determine which face region(s) is/are in the image.
    – For each detected face, use a landmark detector to predict the 68 landmark locations within that face region.
Eye Region Extraction:
  • Extract the eye regions R_left and R_right from the face image and landmarks:
    – Define key landmark indices:
      ∗ Left eye landmarks: left_eye_indices = [37, 38, 39, 40, 41, 42]
      ∗ Right eye landmarks: right_eye_indices = [43, 44, 45, 46, 47, 48]
For the left eye:
min_x_left = min(landmark_x[37:42])
max_x_left = max(landmark_x[37:42])
min_y_left = min(landmark_y[37:42])
max_y_left = max(landmark_y[37:42])
For the right eye:
min_x_right = min(landmark_x[43:48])
max_x_right = max(landmark_x[43:48])
min_y_right = min(landmark_y[43:48])
max_y_right = max(landmark_y[43:48])
For the left eye region:
  • R_left = image[min_y_left:max_y_left, min_x_left:max_x_left]
For the right eye region:
  • R_right = image[min_y_right:max_y_right, min_x_right:max_x_rightt]
```

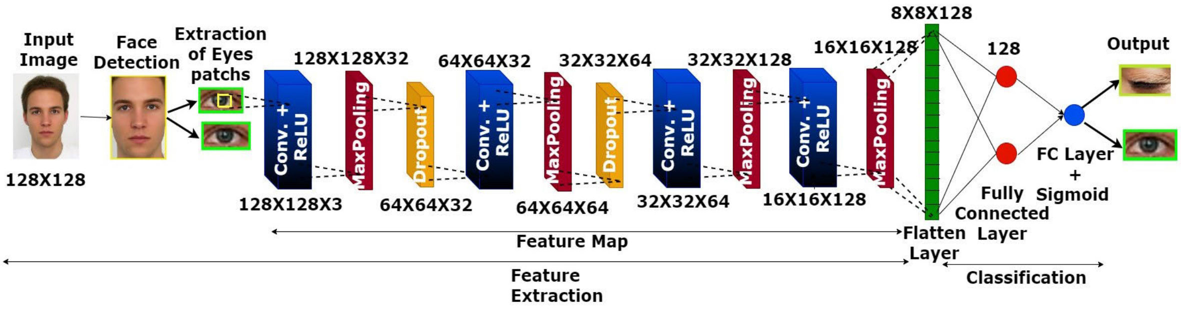


FIGURE 4. Proposed shallow CNN model.

layer is supplied to the third convolution layer of one-twenty-eight 3 × 3 filters, yielding one-twenty-eight 32 × 32 feature maps. The third 2 × 2 max-pooling operations downsize the produced feature maps to one-twenty-eight 16 × 16 feature maps. The last convolution layer of one twenty-eight 1 × 1 filters (with unitary strides), yielding one-twenty-eight 16 × 16 feature maps, which is then downsized to one-twenty-eight 8 × 8 feature maps using the last max-pooling operation of size 2 × 2.

The ReLU activation function applies across convolution blocks to execute the nonlinear process. When the input value is negative, it outputs a null value, which is represented by the equation(3):

$$H(z) = \begin{cases} 0 & z < 0 \\ z & z >= 0 \end{cases} \tag{3}$$

Here ‘z’ is the function’s input. The last two levels of the shallow CNN consist of fully-connected layers. Sigmoid and

Softmax are two of the most commonly used activation functions, the final fully-connected layer. The outcome of fully-connected layer, denoted as y, is computed as equation(4):

$$y = f(W \times x + b) \tag{4}$$

where:

- The weighted total of the inputs is denoted by W × x, where the weight assigned to each input is multiplied.
- b signifies bias.
- f signifies an activation function

The sigmoid activation function, which is referred to as the logistic function, is widely used in convolution neural networks. When solving binary classification issues, where the goal is to reach a binary conclusion, it is typically utilized. Here's the mathematical representation of the sigmoid function in equation(5):

$$\sigma(y) = \frac{1}{1+e^{-y}} \tag{5}$$

where y denotes the Sigmoid function's input, which can be any real number and the natural logarithm's base is e. As seen in *FIGURE 4*, the suggested system's 12th layer presents the aforementioned procedure. The tenth layer's feature maps are flattened and then run through two fully connected layers with 128 nodes. Lastly, the output layer's Sigmoid activation function regulates whether the eyelids are closed or open.

D. TRANSFER LEARNING MODELS FOR DROWSINESS DETECTION

1) VGG NETWORK

Especially in image classification, there are various uses for the popular neural network architecture known as VGG (Visual Geometry Group). VGG-19 is a variant of the VGG network, which in short consists of 19 layers. The construction of VGG-19 is shown in FIGURE 5. The five blocks that make up VGG-19 have 16 convolution layers. Following each convolutional block is a Max-pool layer that minimizes the input image's size by two while increasing several filters in the convolution layer by two. Three dense layers, each measuring 4096, 4096, and 1000 pixels, make up Block 6. Using VGG, the input photos are divided into 1000 unique groups. The dimension of fully-Connection layer 8 is set to two in this study; since there are two output classes.

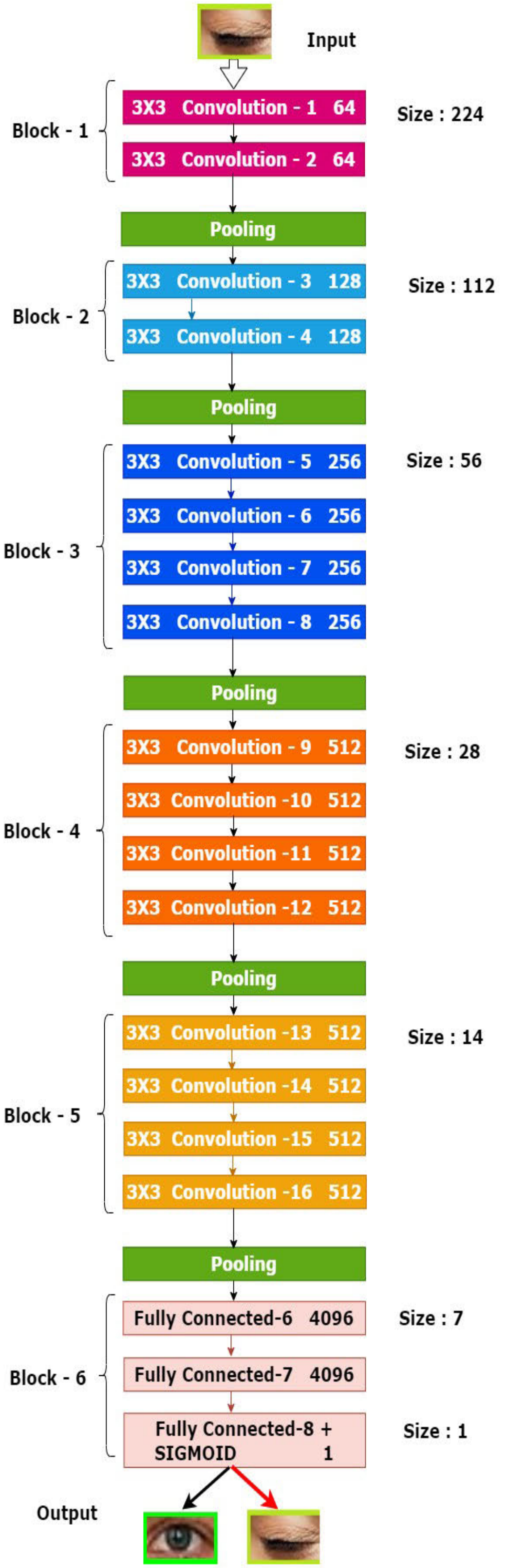


FIGURE 5. [35] VGG-19 architecture.

2) MobileNetV2 MODEL

A new CNN layer called the inverted residual and linear bottleneck layer is included in MobileNetV2, allowing for excellent performance in embedded and mobile vision applications. This new layer serves as the foundation for the MobileNetV2 network, which may be customized to carry out semantic segmentation, item classification, and detection. 19 leftover bottleneck layers are positioned after the first fully-convolution layer, which has 32 filters in the overall design of MobileNetV2. The basis of MobileNetV2 is an inverted residual structure made up of three layers in order:

- A 1 × 1 convolution to expand some channels.
- A depth-wise separable convolution.
- A 1 × 1 convolution to return several channels to its initial value.

MobileNetV2 also uses a technique called linear bottleneck convolutions. This involves using a 1 × 1 convolution without any nonlinearity at the end of the bottleneck layer. This lowers the total parameters and computations required while maintaining the network's accuracy. The construction of MobileNet-V2 is depicted in the FIGURE 6.

3) ResNet50 MODEL

The ResNet-50 architecture is displayed in FIGURE 7. It has one max-pool layer, one average pool layer, and forty-eight convolutional layers. An artificial neural network (ANN) that builds networks by stacking leftover blocks is called a residual neural network. The 50-layer Reset's building block has a bottleneck-like architecture. A bottleneck residual block reduces matrix multiplications and parameter counts by employing 1 × 1 convolutions, often known as a bottleneck. This trains each layer considerably fast.

Skip connections in a residual neural network, which run parallel to the convolutional layers, aid in the network's comprehension of global features. After a few weight levels, the shortcut connection is connected to the output to add the input x (FIGURE 8). The network can optimize many layers for faster training through these shortcut connections by eliminating training on unneeded levels. In terms of mathematics, the output H(x) is defined as equation(6):

$$H(X) = F(X) + X \tag{6}$$

Weight layers are designed to acquire a specific type of residual mapping, denoted by equation(7):

$$F(X) = H(X) - X \tag{7}$$

and the non-linear weight layers stacked are represented by F(x).

4) InceptionV3 MODEL

The InceptionV3 architecture depicted in *FIGURE 9* is based on a series of Inception modules. This enables the network to learn characteristics at various spatial resolutions and scales.

The InceptionV3 architecture consists of the following blocks:

- Stem block: This block uses several convolution and pooling layers to shrink the input image to 32×32 pixels.
- Inception blocks: The InceptionV3 architecture contains nine Inception blocks, which are arranged in a sequential order. Each Inception block consists of four parallel convolution layers with different kernel sizes: 1 × 1, 3 × 3, 5 × 5, and a pooling layer.
- Reduction blocks: The InceptionV3 architecture contains two Reduction blocks, which are employed to lower the feature maps' spatial resolution.
- Auxiliary classifier: The InceptionV3 architecture contains an auxiliary classifier, which is trained to predict image labels from feature maps of an intermediate layer.
- Global average pooling: The InceptionV3 architecture combines feature maps into a single vector representation using global average pooling. GAP is often used to create a fixed-size output for fully connected layers or classifiers, making it suitable for tasks requiring a fixed-length output.
- Fully-connected layer: A fully-connected layer, the last layer in the InceptionV3 architecture, categorizes the image.

E. DATASETS

In this study, we utilize two datasets for evaluating the proposed methodology. The first dataset, Dataset-1, was created by the authors Chirra et al. [34]. This dataset consists of 324 images categorized into open and closed eyes. These images were specifically collected to detect drowsiness based on eye state. The dataset is used to assess the performance of our proposed approach, as illustrated in Figure 10.

On the other hand, Dataset-2 was obtained from Kaggle and is named "yawn_eye_dataset_new" [40]. This dataset comprises 1452 images categorized into open and closed eyes. Dataset-2 was compiled to address drowsiness detection based on eye state, as illustrated in Figure 11. These datasets provide a diverse set of eye images, enabling thorough evaluation and validation of our proposed methodology.

The current datasets, consisting of well-lit RGB images, only partially represent nighttime driving scenarios. This model's primary feasibility is its potential integration into driver monitoring systems for commercial and personal vehicle drivers, providing timely alerts to prevent accidents.

The datasets were split into training, validation, and testing sets as shown in Table 1.

TABLE 1. Datasets splitting.

SNO	Dataset	Classi-fication	Training 48%	Vali-dation 12%	Testing 40%	total
1	Dataset-1	Closed	82	21	69	172
		Open	73	18	61	152
		Total	155	39	130	324
2	Dataset-2	Closed	348	87	291	726
		Open	348	87	291	726
		Total	696	174	582	1452

IV. EXPERIMENTAL RESULT AND ANALYSIS

In this work, two datasets were taken to perform experiments. One dataset contains only eye images, another dataset contains face images. From those face images, first identified the face and then detected the eye area, cropped both eyes separately using a 68-point face landmarks detector

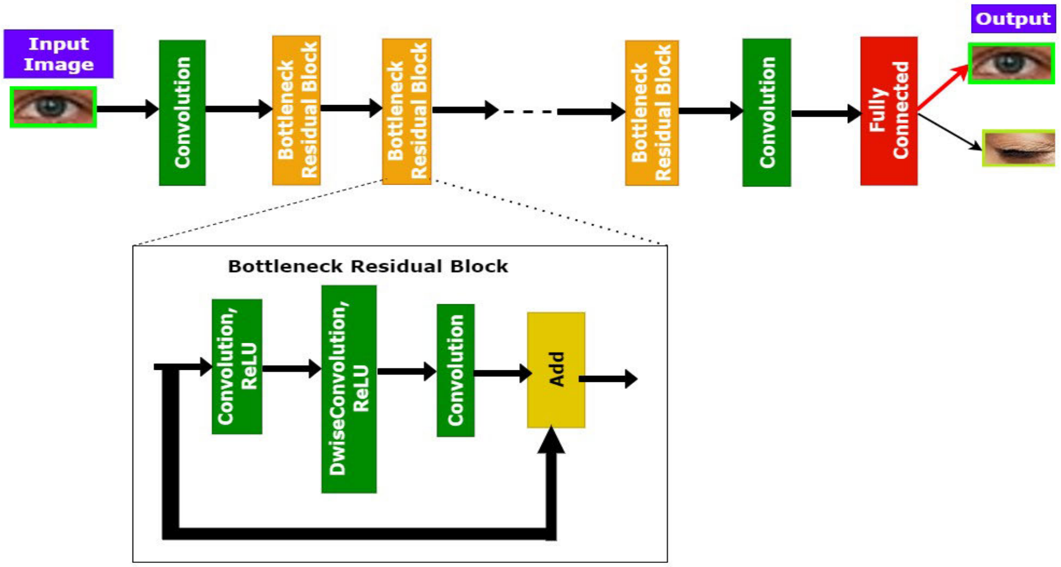


FIGURE 6. [36] MobileNetV2 architecture.

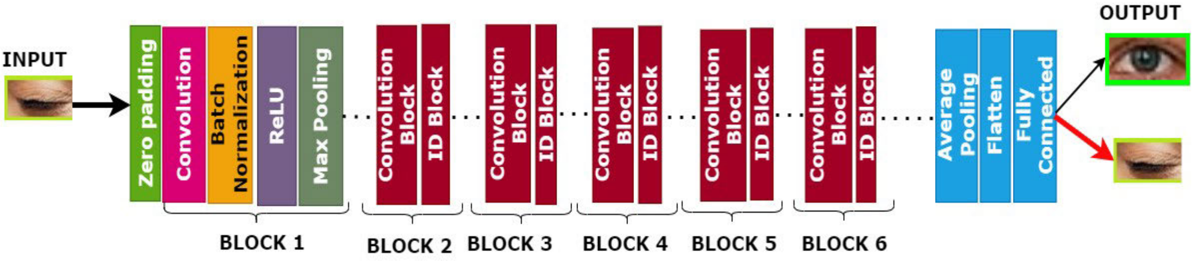


FIGURE 7. [37] ResNet 50 architecture.

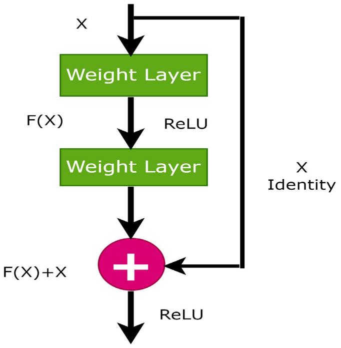


FIGURE 8. [37] Skip connection.

algorithm. There are training and testing categories within this dataset. 40% of the images were taken for testing, while the remaining 60% were part of the training samples. The dataset was available in two categories: closed and open eyes. In FIGURE 10, the sample images are displayed.

A. PROPOSED SHALLOW CNN HYPERPARAMETER SELECTION

A CNN's settings must be adjusted for optimum performance. Some important parameters are batch size, which influences generalization and training speed; convergence behavior is influenced by optimizers like Adam, SGD, Adagrad, etc.; multiple epochs, which shows that the neural network is trained by running the complete dataset multiple times; and learning rate, which determines the optimization step size. The suggested shallow CNN model needs to be adjusted, which requires experimenting with these settings.

1) EFFECTS OF LEARNING RATE

One significant element influencing the CNN model's efficiency is the learning rate. While the loss function gradually decreases with a lower learning rate, a greater learning rate expedites the learning process and raises it. To minimize the cost function in the sleepiness detection classification problem, the ideal learning rate must be chosen. Training of the proposed model with varying learning rates of 0.1, 0.01, 0.001, and 0.0001 was conducted. FIGURE 11 displays the accuracy rate for different learning rates. Depending on the findings, setting the learning rate to 0.001 produced results with higher classification accuracy. The model's reduced learning rate prevents over-fitting by progressively lower errors.

2) EFFECTS OF EPOCHS

It was determined how many epochs yielded the best results in classification accuracy. Training was done on the suggested

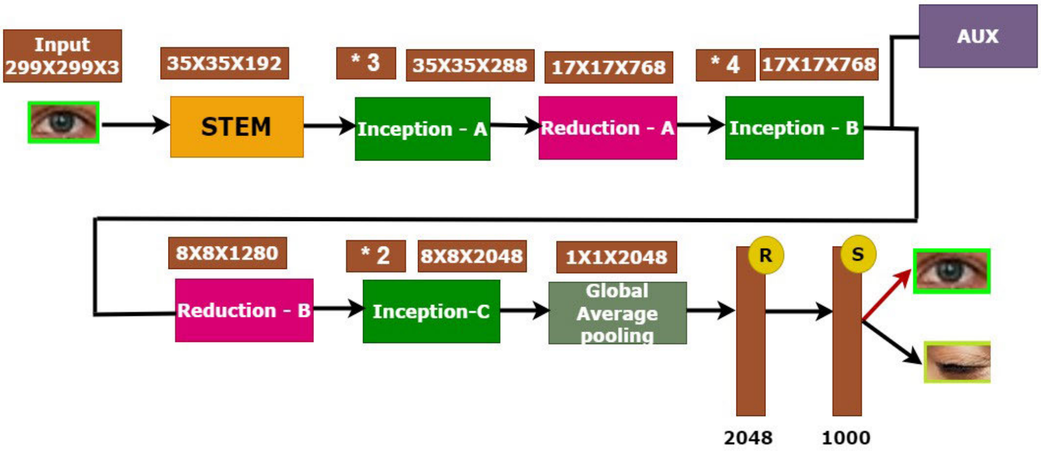
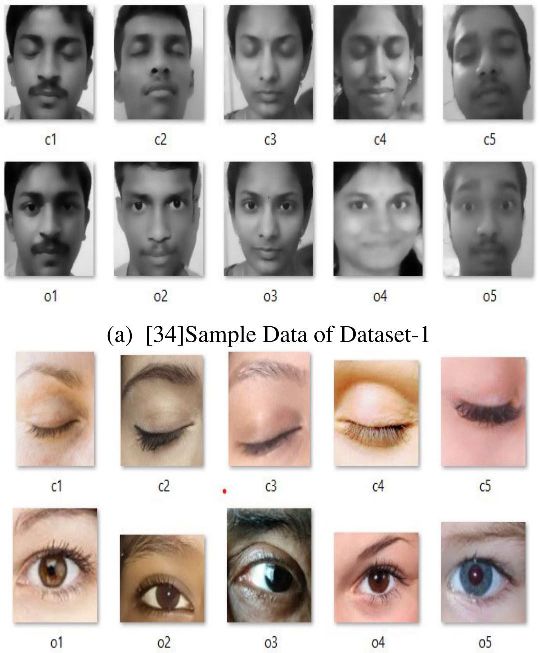


FIGURE 9. [38] Inception-V3 architecture.



(a) [34]Sample Data of Dataset-1

(b) Sample Data of Dataset-2

FIGURE 10. Dataset-1 and Dataset-2 sample images.

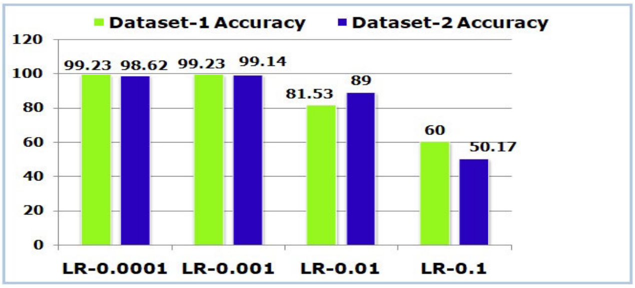


FIGURE 11. Variations in accuracy based on learning rates.

shallow CNN model over 10, 25, 50, and 100 epochs. FIGURE 12 illustrates that both datasets' categorization accuracy was high at 100 epochs. Accuracy performance is enhanced by a larger number of epochs. As a result, 100 was chosen as the ideal for several epochs.

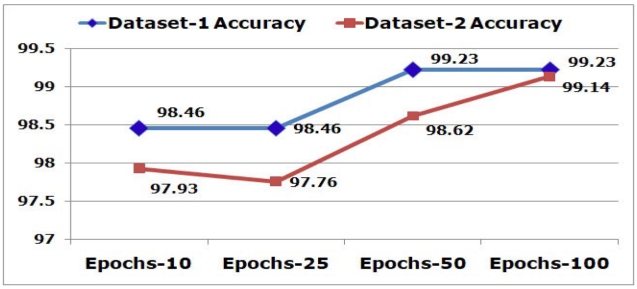


FIGURE 12. Variations in accuracy based on epochs.

3) EFFECTS OF BATCH SIZE

One significant factor that affects the model's accuracy of classification is the batch size. Because of the longer running time and constant weights caused by the larger batch size, the model performs less well overall and uses more memory. As a result, to raise the model quality, the correct batch size is chosen. Evaluations of the suggested model are conducted using batch sizes of 4, 8, 16, and 32. FIGURE 13 compares the model performance for two datasets with varying batch sizes. At 0.0001, the learning rate was implemented for 100 training epochs in the model. According to the results of the experiment, a batch size of 32 is used to train the model to improve final accuracy.

4) EFFECTS OF OPTIMIZER

In deep learning, the optimizer's job is to lower the cost function by updating the bias and weight parameters. By altering the bias and weight values of the model, an optimizer for the problem is selected, leading to faster and better outcomes. The proposed model was assessed using the optimizers RMSprop, Adam, Adagrad, Adadelta, and SGD (Stochastic Gradient Descent). FIGURE 14 displays the model's performance using different optimizer techniques on

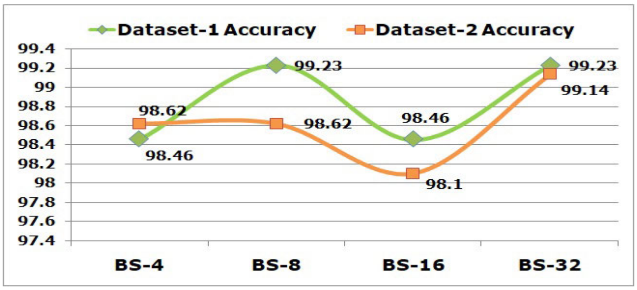


FIGURE 13. Variations in accuracy based on batch size.

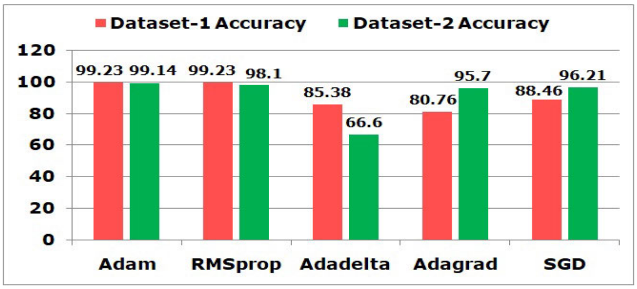


FIGURE 14. Variations in accuracy based on optimizers.

TABLE 2. Evaluation table for test data.

SNO	Dataset	Eye State	Precision	Recall	F1 Score	Accuracy %
1	Dataset-1	Closed	0.99	1.0	0.99	99.23
		Open	1.00	0.98	0.99	
2	Dataset-2	Closed	0.99	0.99	0.99	99.14
		Open	0.99	0.99	0.99	

two datasets. When compared to other optimizer techniques, the shallow CNN model's accuracy increased when using the Adam optimizer.

B. PROPOSED SHALLOW CNN MODEL'S OVERALL PERFORMANCE

A key assessment in the methodology is classification accuracy, which is provided by equation(8):

$$Accuracy = \frac{TN + TP}{TN + FN + TP + FP} \times 100 \tag{8}$$

where FN means False Negative, FP means False Positive, TN is for True Negative, and TP is for True Positive.

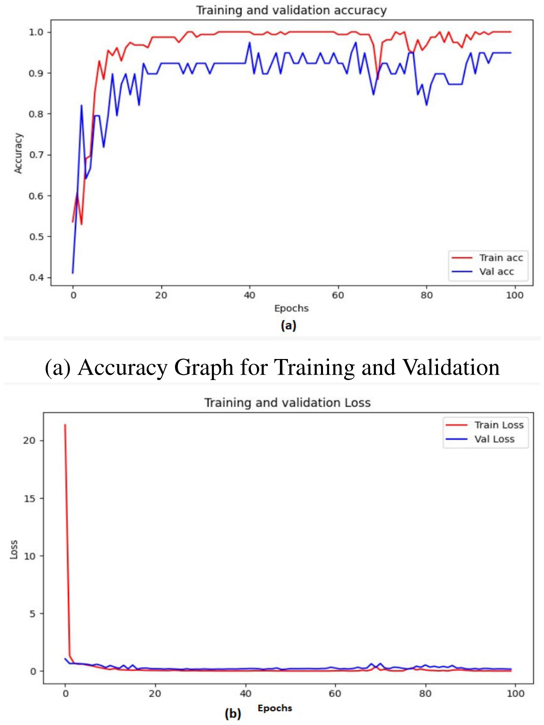
According to the suggested model, closed eyes were classified as negative and open eyes as positive.

The precision was used to calculate the classification's correctness by using the equation(9):

$$Precision = \frac{TP}{FP + TP} \times 100 \tag{9}$$

The effectiveness of classification was calculated using the recall, which is provided by equation(10):

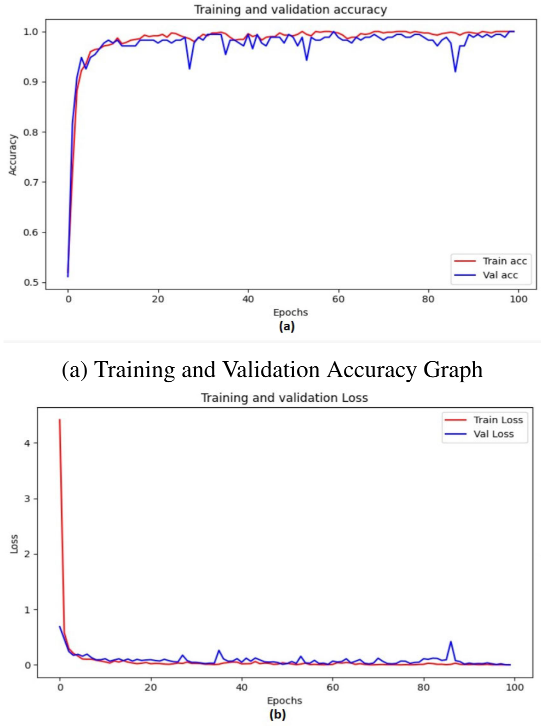
$$Recall = \frac{TP}{FN + TP} \times 100 \tag{10}$$



(a) Accuracy Graph for Training and Validation

(b) Loss Graph for Training and Validation

FIGURE 15. Dataset-1 training and validation graphs.



(a) Training and Validation Accuracy Graph

(b) Training and Validation Loss Graph

FIGURE 16. Dataset-2 training and validation graphs.

Finally, the F1-score was computed by applying the equation(11):

$$F1\_Score = \frac{2 \times Recall \times Precision}{Recall + Precision} \times 100 \tag{11}$$

TABLE 3. Accuracy (%) comparison: proposed shallow CNN vs. deep learning algorithms on both datasets.

SNO	Model	Epochs	Batch Size	Learning Rate	Optimizer	Dataset-1 Accuracy	Dataset-2 Accuracy
1	Inception-V3	100	32	0.001	Adam	96.15	93.12
2	MobileNet-V2	100	32	0.001	Adam	97.69	91.06
3	ResNet-50	100	32	0.001	Adam	98.46	98.1
4	VGG19	100	32	0.001	Adam	98.46	98.62
5	Proposed shallow CNN	100	32	0.001	Adam	99.23	99.14

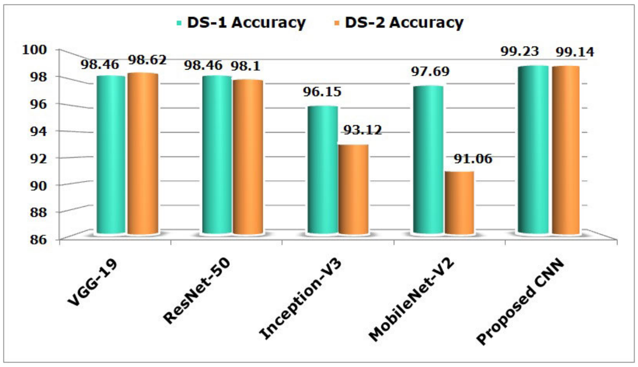
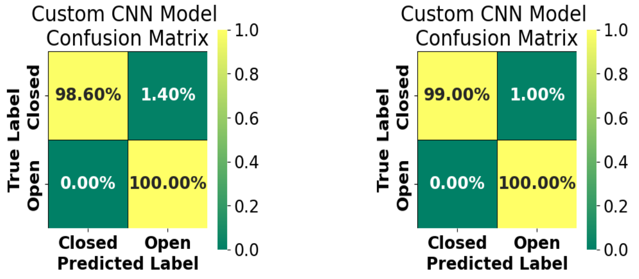


FIGURE 17. Accuracy comparison of various deep learning models and Proposed shallow CNN Model of both Datasets.



(a) Dataset-1 Confusion Matrix

(b) Dataset-2 Confusion Matrix

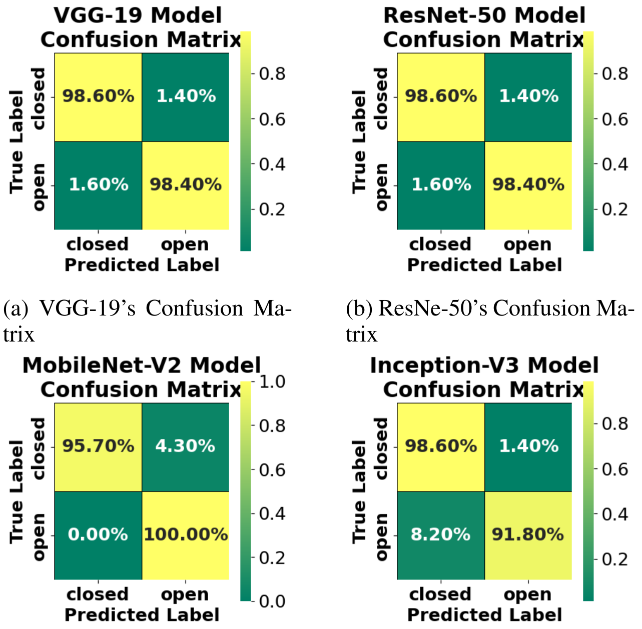
FIGURE 18. Confusion matrices of two Datasets for proposed model.

In binary classification, the output layer usually uses the Sigmoid activation operation, while the cost function is binary cross-entropy. Equation (12) provides the formula for the binary cross-entropy cost function.

$$L(a, \hat{a}) = -[a \times log(\hat{a}) + (1 - a) \times log(1 - \hat{a})] \tag{12}$$

In this study, an evaluation of the proposed shallow CNN model on Dataset-1 and Dataset-2 is presented. FIGURE 15(a) and FIGURE 15(b) displayed loss and accuracy graphs of the proposed model for Dataset-1, while FIGURE 16(a) and FIGURE 16(b) provided the same visualization specifically for Dataset-2.

Table 2 presents performance metrics for a binary classification task involving eye state detection, categorized into "Closed" and "Open" states, in two distinct datasets(Dataset-1 and Dataset-2). The measures, which are expressed as percentages, include recall, precision, F1-score, and total accuracy.



(a) VGG-19's Confusion Matrix

(b) ResNe-50's Confusion Matrix

(c) Confusion Matrix for MobileNet-V2

(d) Confusion Matrix for Inception-V3

FIGURE 19. Confusion matrices for Dataset-1.

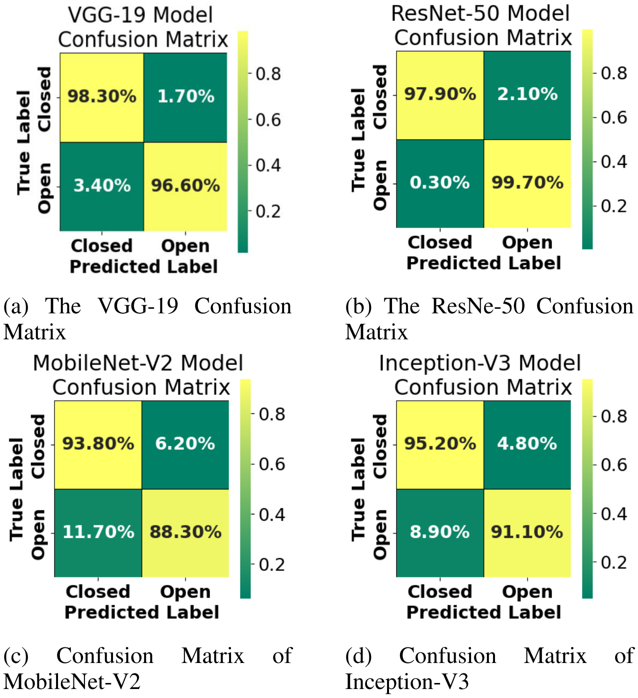
The classifier obtained high recall (1.0), F1-score (0.99), and precision (0.99) for the "Closed" condition, yielding an accuracy of 99.23%. Similarly, the classifier showed perfect precision (1.0), good recall (0.98), and an F1-score of 0.99 for the "Open" condition, resulting in an accuracy of 99.23%. For both the "Closed" and "Open" states, the classifier maintained high recall(0.99) and precision(0.99), yielding F1-scores of 0.99 for both classes. On Dataset-2, the combined accuracy for both classes was 99.14%.

These findings imply that the suggested shallow CNN model performed exceptionally well on both datasets, achieving high precision, recall, and F1 scores, which ultimately translated into impressive accuracy rates. The robustness of the classifier across different datasets indicates its effectiveness in accurately detecting eye states, making it drowsiness detection.

Table 3 provides a comparative analysis of different CNN models trained on datasets to identify the drowsiness of the driver, showcasing their performance in terms of accuracy

TABLE 4. Comparing different approaches for detecting drowsiness.

SNO	Reference	Method	ROI	Dataset	Accuracy(%)
1	T. Faisal, et al. [15]	CNN Model	Eyes	Own dataset	97.98
2	B. Ganguly, et al. [16]	f-RCNN	Eye	Blink analysis and angular views (60 degrees right- left) of eyes dataset	97.6
3	I. Jahan, et al. [19]	Custom CNN 4D	Eye	MRL Eye dataset	97.53
4	B. Akrout, et al. [20]	MobileNet-V2, LSTM	Eyes and Mouth	MiraclHB, YawDD, and DEAP	98.4
5	V. Kumar, et al. [21]	Inception-V3 +LSTM	Eyes & Mouth	NTHU-DDD	93.69
6	T. Abbas, et al. [26]	ReSVM	Face	State Farm Distracted Driver Detection, Boston University, DrivFace, and FT-UMT.	95.5
7	H. Jia, et al. [27]	MTCNN	Mouth and Eyes	WIDER FACE and MTFL datasets	97.5
8	S. Jamshidi, et al. [30]	HDDD+LSTM	Mouth & Eyes	NTHU-DDD	87.19
9	S. Saurav, et al. [31]	DCNNE	Eye	ZJU, CEW, and MRL	97.99
10	V. R. R. Chirra, et al. [34]	Haar Cascade	Eyes	Own dataset	96.42
11	proposed Model	shallow CNN	Eyes	Dataset-1 [34] and Dataset-2 [40]	99.23, 99.14



(a) The VGG-19 Confusion Matrix

(b) The ResNe-50 Confusion Matrix

(c) Confusion Matrix of MobileNet-V2

(d) Confusion Matrix of Inception-V3

FIGURE 20. Confusion matrices for Dataset-2.

on two separate datasets(Dataset-1 and Dataset-2) across various hyperparameters like epochs, batch size, optimizer, and learning rate.

The CNN architectures utilized in this study, include Inception-V3, MobileNet-V2, ResNet-50, VGG19, and a proposed shallow CNN model. The Adam optimizer was used for all models.Inception-V3 model attained an accuracy of 96.15% on Dataset-1 and 93.12% on Dataset-2.MobileNet-V2 demonstrated improved performance with an accuracy of 97.69% on Dataset-1 but a lower accuracy of 91.06% on Dataset-2. ResNet-50 model displayed high accuracy on both datasets, with 98.46% on Dataset-1 and 98.1% on Dataset-2. VGG19 model yielded similar accuracy to ResNet-50, achieving 98.46% on Dataset-1 and slightly higher accuracy of 98.62% on Dataset-2. The Proposed Shallow CNN model outperformed all other models, achieving the highest accuracies of 99.23% on Dataset-1 and 99.14% on Dataset-2.

In summary, the proposed shallow CNN model demonstrated superior performance compared to pre-trained CNN architectures such as Inception-V3, MobileNet-V2, ResNet-50, and VGG19, on both datasets. These outcomes highlight the usefulness of the suggested model architecture when it comes to driver sleepiness detection datasets that are assessed based on eye state.

The accuracy attained by five distinct CNN architectures on two distinct datasets is displayed in FIGURE 17. The models were trained using the Adam optimizer for 100 epochs with a learning rate of 0.001 and a batch size of 32.

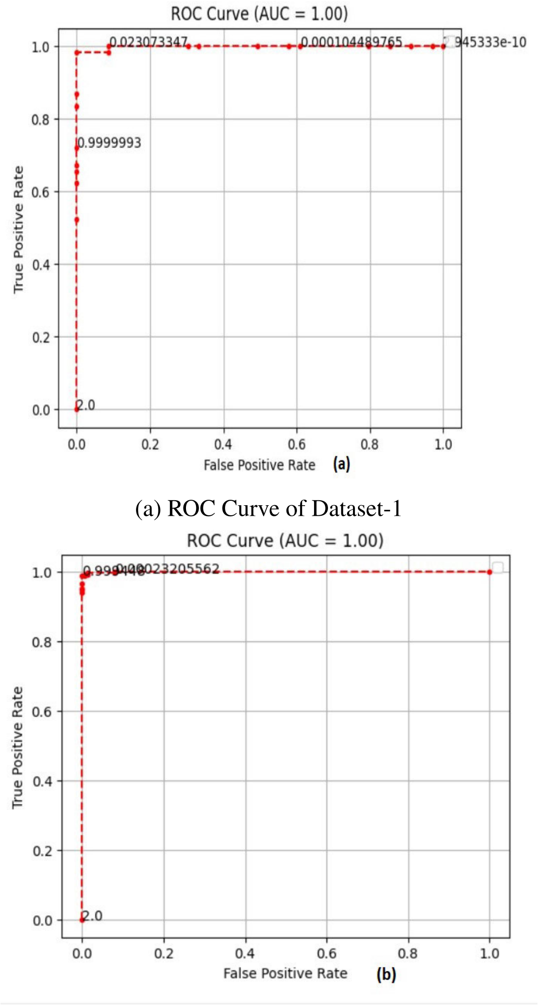
The x-axis of the graph lists the five CNN architectures: MobileNet-V2, VGG-19, ResNet-50, Inception-V3, and the proposed shallow CNN. The y-axis represents the accuracy achieved by each model, expressed as a percentage.

According to FIGURE 17, the proposed shallow CNN model outperformed all other models on both datasets, obtaining an accuracy of 99.23% on Dataset-1 and 99.14% on Dataset-2. VGG-19 and ResNet-50 achieved similar accuracy, with 98.46% and 98.10% on Dataset-1, respectively. Inception-V3 and MobileNet-V2 had lower accuracy on Dataset-2, at 93.12% and 91.06%, respectively.

Overall, the suggested shallow CNN model is a more effective architecture for determining drowsiness detection.

The confusion matrices of proposed shallow CNN are depicted in FIGURE 18(a) and FIGURE 18(b). The pre-trained models confusion matrices are shown in FIGURE 19 and FIGURE 20.

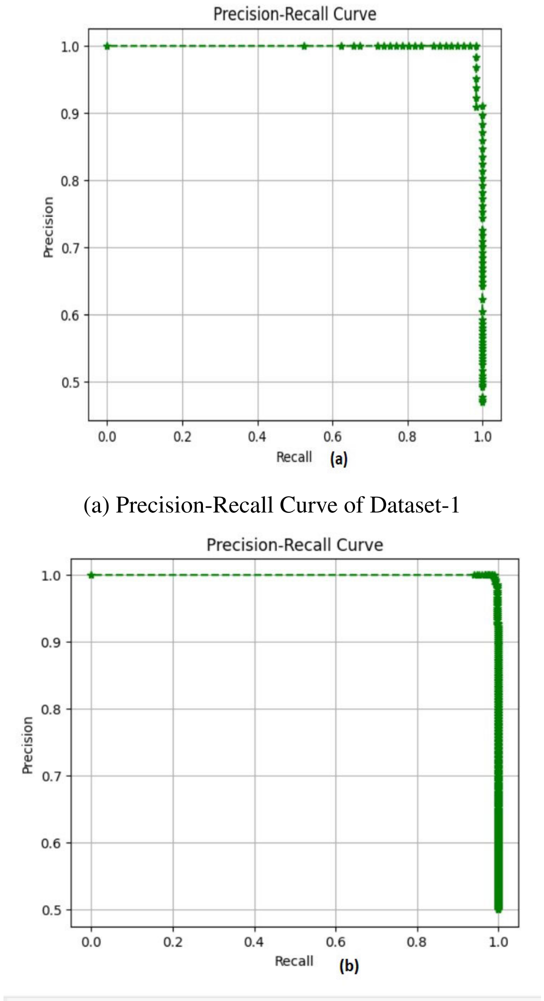
For both Dataset-1 and Dataset-2, the ROC curves are displayed for the proposed shallow CNN in FIGURE 21.



(a) ROC Curve of Dataset-1

(b) ROC Curve of Dataset-2

FIGURE 21. **Receiver operating characteristic(ROC) curves.**



(a) Precision-Recall Curve of Dataset-1

(b) Precision-Recall Curve of Dataset-2

FIGURE 22. **Precision-recall curves(PRC).**

At the same time, the PR Curves for both datasets can be found in FIGURE 22 respectively.

A thorough comparison of the face techniques applied in the region of interest (ROI) for drowsiness recognition in other relevant publications is shown in Table 4. The suggested approach achieved an accuracy better than 99% in these studies, while the other approaches' accuracy varied from 87.19% to 98.4%. Some methods [20], [21], [27] and [30] focused on both the eyes and the mouth, while one approach [26] focused on the entire face to identify drowsiness. The methods employed by [15], [16], [19], [31] and [34] concentrated on the eyes.

While models like Inception V3 and MobileNet V2 are well-established and widely used, the novelty of this study lies in the proposed shallow CNN model, which used Dlib's 68-point facial landmarks to find eye regions and to determine whether the eyes are open or close. The shallow CNN model employed fewer layers than the pre-trained models. However, the shallow CNN model got high accuracy and was computationally efficient.

C. DISCUSSION

This study compared the drowsiness detection results of the pre-trained models VGG-19, ResNet50, MobileNetV2, and InceptionV3 with those of a shallow CNN architecture. With fewer layers, the shallow CNN focuses on extracting key visual information like eyelid closure. It offers speed, simplicity, and a lower risk of overfitting, making it effective with limited training data. This ensures excellent accuracy and quick processing of recognized drowsiness-related characteristics from facial landmarks.

However, the generalizability of our results is limited due to the small size and lack of diversity in the datasets used. The study's accuracy, while commendable, raises concerns about robustness in real-world scenarios. Future research needs to address significant limitations by incorporating larger and more diverse datasets that account for variations in lighting conditions, ethnicities, and head poses. This will help validate the model's performance across different environments and populations. The discussion will highlight these constraints and potential biases to provide a clearer understanding of the study's limitations.

The inference time of our shallow CNN model with several pre-trained models. For Dataset-1, inference times were: shallow CNN (1430.141 ms), VGG19 (1851.106ms), ResNet50 (2494.112 ms), MobileNetV2 (2897.924 ms), and InceptionV3 (3233.497 ms). For Dataset-2, times were: shallow CNN (164.497 ms), VGG19 (390.445 ms), ResNet50 (3794.743 ms), MobileNetV2 (1547.561 ms), and InceptionV3 (7229.007 ms). These results show that the proposed shallow CNN model has significantly lower inference times, making it more suitable for real-time driver drowsiness detection compared to more complex pre-trained models.

V. CONCLUSION

This research explored the effectiveness of utilizing a shallow Convolutional Neural Network (CNN) for driver drowsiness identification with limited training data by focusing on key visual cues like eye closure. To address the challenge of drowsiness classification, pre-trained models such as MobileNetV2, ResNet-50, InceptionV3, and VGG-19 were leveraged, and their results were compared with the proposed model. The impact of hyperparameters was also examined. The findings show that, with its suitable architecture, the suggested shallow CNN model surpasses pre-trained models in classifying eye states, achieving classification accuracies of 99.23% and 99.14% on Dataset-1 and Dataset-2, respectively. This research underscores the advantages of employing a shallow CNN: reduced data dependency, lightweight architecture, and efficient computation. However, limitations remain, such as the need for a more diverse dataset and the potential for overfitting due to the small size of the data. Future work will involve training the model with infrared or near-infrared images for better performance in low-light conditions and exploring diverse datasets, including nighttime and adverse weather scenarios. This will enhance the model's real-world applicability, ensuring it can benefit a wide range of drivers by improving road safety and reducing accidents caused by driver fatigue. Also, it explores integrating additional modalities like physiological sensors or EEG data and incorporating yawning or head pose to enrich the feature space and improve detection accuracy. Personalized drowsiness models that adapt to individual driver characteristics could also enhance safety interventions.

ACKNOWLEDGMENT

The authors express their gratitude to VIT-AP University for its support and provision of resources that facilitated the execution of this research.

CONFLICTS OF INTEREST

"The authors declare no conflict of interest."

REFERENCES

[1] H. Summala, "Towards understanding motivational and emotional factors in driver behaviour: Comfort through satisficing," in *Modelling Driver Behaviour in Automotive Environments: Critical Issues in Driver Interactions With Intelligent Transport Systems*. London, U.K.: Springer, 2007, pp. 189–207.

[2] T. Igasaki, K. Nagasawa, N. Murayama, and Z. Hu, "Drowsiness estimation under driving environment by heart rate variability and/or breathing rate variability with logistic regression analysis," in *Proc. 8th Int. Conf. Biomed. Eng. Informat. (BMEI)*, Oct. 2015, pp. 189–193.

[3] M. Papakostas, K. Das, M. Abouelenien, R. Mihalcea, and M. Burzo, "Distracted and drowsy driving modeling using deep physiological representations and multitask learning," *Appl. Sci.*, vol. 11, no. 1, p. 88, Dec. 2020.

[4] M. Awais, N. Badruddin, and M. Drieberg, "A hybrid approach to detect driver drowsiness utilizing physiological signals to improve system performance and wearability," *Sensors*, vol. 17, no. 9, p. 1991, Aug. 2017.

[5] M. Ramzan, H. U. Khan, S. M. Awan, A. Ismail, M. Ilyas, and A. Mahmood, "A survey on state-of-the-art drowsiness detection techniques," *IEEE Access*, vol. 7, pp. 61904–61919, 2019.

[6] Zebra. (Jan. 2023). *Drowsy Driving Statistics*. [Online]. Available: https://www.thezebra.com/resources/research/drowsy-driving-statistics/

[7] M. M. Ahsan, Y. Li, J. Zhang, M. T. Ahad, and M. M. S. Yazdan, "Face recognition in an unconstrained and real-time environment using novel BMC-LBPH methods incorporates with DJI vision sensor," *J. Sensor Actuator Netw.*, vol. 9, no. 4, p. 54, Nov. 2020.

[8] A. Liu, Z. Li, L. Wang, and Y. Zhao, "A practical driver fatigue detection algorithm based on eye state," in *Proc. Asia–Pacific Conf. Postgraduate Res. Microelectron. Electron. (PrimeAsia)*, Sep. 2010, pp. 235–238.

[9] A. Królak and P. Strumiłło, "Eye-blink detection system for human–computer interaction," *Universal Access Inf. Soc.*, vol. 11, no. 4, pp. 409–419, Nov. 2012.

[10] Z.-T. Liu, C.-S. Jiang, S.-H. Li, M. Wu, W.-H. Cao, and M. Hao, "Eye state detection based on weight binarization convolution neural network and transfer learning," *Appl. Soft Comput.*, vol. 109, Sep. 2021, Art. no. 107565.

[11] S. Fuangkaew and K. Patanukhom, "Eye state detection and eye sequence classification for paralyzed patient interaction," in *Proc. 2nd IAPR Asian Conf. Pattern Recognit.*, Nov. 2013, pp. 376–380.

[12] M. A. R. Ahad, S. Kobashi, and J. M. R. Tavares, "Advancements of image processing and vision in healthcare," *J. Healthcare Eng.*, vol. 2018, Jan. 2018, Art. no. 8458024.

[13] A. Koesdwiady, R. Soua, F. Karray, and M. S. Kamel, "Recent trends in driver safety monitoring systems: State of the art and challenges," *IEEE Trans. Veh. Technol.*, vol. 66, no. 6, pp. 4550–4563, Jun. 2017.

[14] A.-C. Phan, T.-N. Trieu, and T.-C. Phan, "Driver drowsiness detection and smart alerting using deep learning and IoT," *Internet Things*, vol. 22, Jul. 2023, Art. no. 100705.

[15] T. Faisal, I. Negassi, G. Goitom, M. Yassin, A. Bashir, and M. Awawdeh, "Systematic development of real-time driver drowsiness detection system using deep learning," *IAES Int. J. Artif. Intell. (IJ-AI)*, vol. 11, no. 1, p. 148, Mar. 2022.

[16] B. Ganguly, D. Dey, and S. Munshi, "An integrated system for drivers' drowsiness detection using deep learning frameworks," in *Proc. IEEE VLSI Device Circuit Syst. (VLSI DCS)*, Feb. 2022, pp. 55–59.

[17] E. M. Lopez, M. P. S. Lorente, J. M. A. Weber, and M. A. S. de Miguel, "Driver drowsiness detection by applying deep learning techniques to sequences of images," *Appl. Sci.*, vol. 12, no. 3, p. 1145, 2022.

[18] R. Florez, F. Palomino-Quispe, R. J. Coaquira-Castillo, J. C. Herrera-Levano, T. Paixão, and A. B. Alvarez, "A CNN-based approach for driver drowsiness detection by real-time eye state identification," *Appl. Sci.*, vol. 13, no. 13, p. 7849, Jul. 2023.

[19] I. Jahan, K. M. A. Uddin, S. A. Murad, M. S. U. Miah, T. Z. Khan, M. Masud, S. Aljahdali, and A. K. Bairagi, "4D: A real-time driver drowsiness detector using deep learning," *Electronics*, vol. 12, no. 1, p. 235, Jan. 2023.

[20] B. Akrout and S. Fakhfakh, "How to prevent drivers before their sleepiness using deep learning-based approach," *Electronics*, vol. 12, no. 4, p. 965, Feb. 2023.

[21] V. Kumar and S. Sharma, "Driver drowsiness detection using modified deep learning architecture," *Evol. Intell.*, vol. 16, no. 6, pp. 1907–1916, Dec. 2023.

[22] P. Liu, H.-L. Chi, X. Li, and J. Guo, "Effects of dataset characteristics on the performance of fatigue detection for crane operators using hybrid deep neural networks," *Autom. Construct.*, vol. 132, Dec. 2021, Art. no. 103901.

[23] Z. Mu, L. Jin, J. Yin, and Q. Wang, "Research on a driver fatigue detection model based on image processing," *Hum.-Centric Comput. Inf. Sci.*, vol. 12, pp. 1–17, Apr. 2022.

[24] A.-C. Phan, N.-H.-Q. Nguyen, T.-N. Trieu, and T.-C. Phan, "An efficient approach for detecting driver drowsiness based on deep learning," *Appl. Sci.*, vol. 11, no. 18, p. 8441, Sep. 2021.

[25] T. Zhu, C. Zhang, T. Wu, Z. Ouyang, H. Li, X. Na, J. Liang, and W. Li, "Research on a real-time driver fatigue detection algorithm based on facial video sequences," *Appl. Sci.*, vol. 12, no. 4, p. 2224, Feb. 2022.

[26] T. Abbas, S. F. Ali, M. A. Mohammed, A. Z. Khan, M. J. Awan, A. Majumdar, and O. Thinnukool, "Deep learning approach based on residual neural network and SVM classifier for driver's distraction detection," *Appl. Sci.*, vol. 12, no. 13, p. 6626, Jun. 2022.

[27] H. Jia, Z. Xiao, and P. Ji, "Fatigue driving detection based on deep learning and multi-index fusion," *IEEE Access*, vol. 9, pp. 147054–147062, 2021.

[28] G. M. Mohamed, S. S. Patel, and N. Naicker, "Data augmentation for deep learning algorithms that perform driver drowsiness detection," *Int. J. Adv. Comput. Sci. Appl.*, vol. 14, no. 1, pp. 674–682, 2023.

[29] M. Dua, R. Singla, S. Raj, and A. Jangra, "Deep CNN models-based ensemble approach to driver drowsiness detection," *Neural Comput. Appl.*, vol. 33, no. 8, pp. 3155–3168, Apr. 2021.

[30] S. Jamshidi, R. Azmi, M. Sharghi, and M. Soryani, "Hierarchical deep neural networks to detect driver drowsiness," *Multimedia Tools Appl.*, vol. 80, no. 10, pp. 16045–16058, Apr. 2021.

[31] S. Saurav, P. Gidde, R. Saini, and S. Singh, "Real-time eye state recognition using dual convolutional neural network ensemble," *J. Real-Time Image Process.*, vol. 19, no. 3, pp. 607–622, Jun. 2022.

[32] J. S. Bajaj, N. Kumar, R. K. Kaushal, H. L. Gururaj, F. Flammini, and R. Natarajan, "System and method for driver drowsiness detection using behavioral and sensor-based physiological measures," *Sensors*, vol. 23, no. 3, p. 1292, Jan. 2023.

[33] J. Flores-Monroy, M. Nakano-Miyatake, G. Sanchez-Perez, and H. Perez-Meana, "Visual-based real time driver drowsiness detection system using CNN," in *Proc. 18th Int. Conf. Electr. Eng., Comput. Sci. Autom. Control (CCE)*, Nov. 2021, pp. 1–5.

[34] V. Chirra, S. ReddyUyyala, and V. Kishorekolli, "Deep CNN: A machine learning approach for driver drowsiness detection based on eye state," *Revue d'Intell. Artificielle*, vol. 33, no. 6, pp. 461–466, Dec. 2019.

[35] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*.

[36] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4510–4520.

[37] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[38] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2818–2826.

[39] D. Li, Z. Wang, Q. Gao, Y. Song, X. Yu, and C. Wang, "Facial expression recognition based on electroencephalogram and facial landmark localization," *Technol. Health Care*, vol. 27, no. 4, pp. 373–387, Jul. 2019.

[40] *Yawn_Eye_Dataset_New*. Accessed: Jul. 10, 2023. [Online]. Available: https://www.kaggle.com/datasets/serenaraju/yawn-eye-dataset-new/data

MADDURI VENKATESWARLU received the master's degree in computer applications from the JKC College, Acharya Nagarjuna University, Guntur, India, and the master's degree in computer science and engineering from the Al-Ameer College of Engineering and IT, Jawaharlal Nehru Technological University, Kakinada, India. His research interests include computer vision, digital image processing, deep learning, and machine learning. His current research interest includes drowsiness detection based on eye state.

VENKATA RAMI REDDY CH received the Ph.D. degree from the Department of Computer Applications, National Institute of Technology, Tiruchirappalli, Tamil Nadu, India. He is currently working as a Senior Assistant Professor with the School of Computer Science and Engineering, VIT-AP University, Andhra Pradesh. He has more than 13 years of teaching experience. He has published more than 33 research articles in reputed international conferences and journals. His current research interests include computer vision, digital image processing, and machine learning.