A CNN-based Deep Learning Framework for Driver's Drowsiness Detection

Ali Sohail[1], Asghar Ali Shah[2], Sheeba Ilyas[3], Nizal Alshammry[4]
Department of Computer Science, Minhaj University, Lahore, Pakistan[1, 3]
Center of Excellence in Artificial Intelligence (CoE-AI)-Department of Computer Science,
Bahria University, Islamabad, 04408, Pakistan[2]
Department of Computer Sciences-Faculty of Computing and Information Technology,
Northern Border University, Rafha 91431, Saudi Arabia[4]

***Abstract*—Accidents are one of the major causes of injuries and deaths worldwide. According to the WHO report, in 2022 an estimated 1.3 million people die from road accidents. Driver fatigue is the primary factor in these traffic accidents. There are a number of studies presented by previous researchers in the context of driver's drowsiness detection. The majority of earlier strategies relied on image processing systems that used algorithms to identify the yawning, eye closure, and eyebrow of the driver taken from the live video camera. One of the major issues of the previous studies was the delay in detection time and dataset. These studies used physical sensors for monitoring the driver's behavior causes in delay time of detection. In this article, a deep learning approach is used to provide a continuous strategy for detecting driver's drowsiness using an efficient dataset. The trained algorithm is employed on the video taken from the live camera to extract the driver's facial landmarks, which are subsequently processed by a trained algorithm to provide results. The dataset used for training the CNN algorithm is consisting of 2904 images taken from various subjects under various driving circumstances. The data is preprocessed by different methods including statistical moments, CNN filters, frequency vector determination and position Incidence vector calculation. After training the algorithm the feature-based cascade classifiers files are used to recognize the face from the real-life scenario using the live camera. The accuracy of the purposed model is 95%, which is the highest of all the purposed models, based on data gathered from different kind of scenarios.**

Keywords*—Drowsiness detection face detection; eye detection; yawn detection; deep learning; convolutional neural network; electroencephalograph; eye aspect ratio

I. Introduction

There are a number of deaths are caused by road accidents every year. Technology plays a vital role in every field of life [1] [2]. Computational and statistical studies are also participating in the scenario of drowsiness detection which is one of the major causes of these accidents. Drowsiness can't be directly seen; therefore, we must make predictions instead. According to the report of WHO each year almost 1.3 million people lost their lives due to road accidents [3]. Traffic accidents are becoming more frequent as a result of drivers' less supervision of their vehicles which is a serious issue in society. The majority of these road accidents are caused by the driver's health, and 30% of them are brought on by driver fatigue. In this scenario, it is very important to use specialized methods to monitor the driver's driving behavior and warn him when they seem to be falling asleep. Lack of sleep, sleep disorders, alcohol use, and continuous driving are some of the main causes of sleepiness. If drowsiness can be predicted, it would undoubtedly save many lives that would otherwise be lost in fatal car accidents. Measurements of driver tiredness and distraction detection are crucial components of a driver monitoring system. Drowsiness detection fundamentally involves tracking a driver's actions, such as their acceleration, braking, steering, and pedal movement. The signs of tiredness in a driver, on the other hand, include eye movement [4], facial expressions [5], heart rate, breathing rate, and brain activity. The most useful element for determining a driver's drowsiness is their facial expressions. The three main methods of determining facial expressions are image processing [6] methods, artificial neural network (ANN) techniques [7], and electroencephalography (EEG) [8] approaches. For the detection of image-based approaches, template match image-processing and yawn-based techniques are also beneficial. These are image-processing-based computer vision algorithms. Mostly used computer vision methods for detecting driver sleepiness use the driver's head motions [9] and facial expressions, such as blinking eyes [10].

The purpose of the proposed study is to develop a state of the art research for detecting the driver's behavior to avoid road accidents. In previous studies there are the problems of continuous face detection. As driver pass from various circumstances and positions while driving. The previous studies fail to detect the drowsiness under various circumstances of face conditions. This study aims to cover the loophole of the delay time in detection using state of the art Deep-CNN approach with an efficient dataset. Most of the studies presented in the past only detect drowsiness. The proposed study detects and alarms the driver if he is drowsy. The proposed research used the Convolutional neural network (CNN) approach to create the drowsiness detection model using a big dataset of images consists of 2904 images of various people under various circumstances of driving behaviors. Eye blink ratio (EAR) is the key factor of detection. The Viola Jone method is utilized for the detection of face in the live camera approach. The video frames of the extracted live camera videos are passed through CNN trained algorithm and system shows a continuous detection without time delay. As well as the system, is not bounded with the hardware (sensors) problems. As in the previous study the sensors are used to detect the driver's face conditions, movements and behavior. The study only used a live camera that is already placed in the steering wheel.

II. Background

In the past, a number of researchers used different methodologies along with different algorithms and datasets for the identification of drivers' drowsiness. This section of research presents some of the latest computational researches proposed for drowsiness detection.

Researchers present an automatic vehicle control system for fatigue detection [11]. This model was designed for early fatigue detection for a train driver. Whenever a driver is sleepy or in an unconscious condition the system determined it by the movement of his head. Heart sensors are used in this procedure to identify tiredness caused by any serious medical conditions. The technologies used in this system include face identification, Matlab, AVR Studio, and image processing. When a user becomes fatigued, the hardware system alerts the microprocessors. The working of the proposed system is shown in Fig. 1[11].

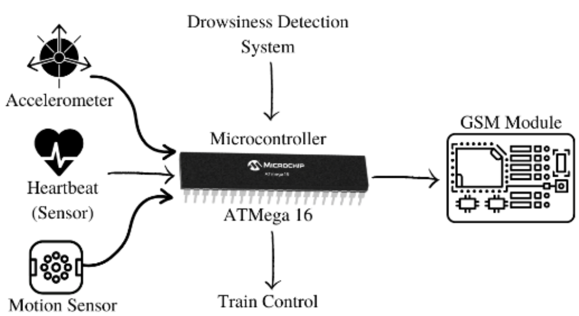

Fig. 1. Block diagram of the proposed system.

Yawn detection [12] plays a vital role in the drowsiness detection. Yawn is the primary indication of drowsiness that is detected by using facial segmentation. In an approach template for Gravity-Center [13] this method is used for facial segment detection. The geometrical arrangements of the mouth and eye are virtually identical. The yawn is measured from the chin to the middle of the nostrils. Grey projection and the Gabor wavelets method [14] were utilized to identify mouth corners. In the last step, the LDA is used to categorize the characteristics to identify yawning as explained in Fig. 2 [12].

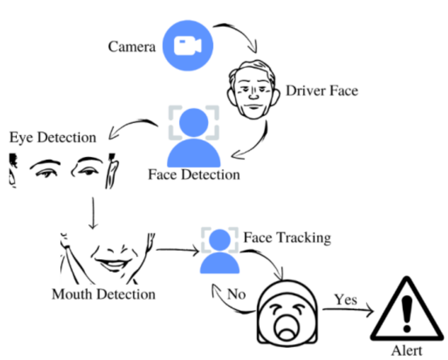

Fig. 2. Drowsiness detection using yawn.

A researcher uses the eye closure period of the driver by recognizing his face in the Eye Based Drowsiness Detection approach [10]. In order to assess the blinking rate, the length of the eye, the location of the iris and eye condition at different time-stamped are examined. Edge detection, a key component of image processing is used for measuring eye aspect ratio. If the eye is closed for 5–6 consecutive frames throughout this procedure, a warning alert is generated. The EAR is calculated by Eq. (1)

$$\text{EAR} = \frac{|P2-P6+|P3-P5|}{2|P1-P4|} \tag{1}$$

Representation Learning is also used to detect the driver's drowsiness using various properties extracted from a dataset [15]. Convolutional neural networks (CNN) are used in this method to capture the most recent facial expressions and challenging non-linear feature interactions [14]. This method involves training a dataset of 30 drivers with a variety of traits under various conditions, such as varied levels of weariness, facial hair, hair fringes, eye size, face shape, and skin tone. The dataset has been separated into five folders, one for training and the other four for validation. The CNN Model trains 50 pictures every cycle. A multi-layer perceptron [16] is employed in this method as a result of the multi-layer categorization methodology. It is most often used for nonlinear issues and classifiers. This method has an accuracy rate of higher than 80%. In the latest research machine learning and deep learning algorithms are used for efficient drowsiness detection. Algorithms for machine learning use supervised, unsupervised, semi-supervised, and reinforcement learning techniques [16]. A method uses several supervised machine learning techniques for the identification of drowsiness. This study used the dataset developed by National Advanced Driving Simulator (NADS-1) [17]. The dataset comprises 27 characteristics, 15,000 tuples, and 144 runs, of which 72 take place during the day and 72 during the night. Participants were asked to operate vehicles on various highways throughout various historical periods. These tests are conducted on the Weka machine learning workbench [18]. In these works, Naive Bayes, Random forest trees, Sequential Minimal Optimization (SMO), and Logistic Regression [19] were employed as machine learning techniques for detection. Two distinct sets of features including Pre Run aggregate feature and Per Event aggregate feature are produced after the data has undergone preprocessing to fill in the missing values [20]. 10-FCV is the testing technique used by the model [21]. Without picking attributes, the SMO produces the best results for the Per Run aggregate features (0.66 F1 scores), however afterward choosing 10 characteristics after an entire of 76, the presentation of the accidental plantation method and Logistic regression improve and provide superior results (0.71 F1 scores). However, for Per Event Aggregate Structures, Logistic based Regression performs best with 0.72 F1 score and a 0.76 ROC area when no features are chosen, whereas SMO performs best with a 0.78 F1 and a 0.78 ROC area when 100 attributes are chosen from a total of 1900 attributes.

Electroencephalogram (EEG) based method is also utilized to find the brain condition for detecting drowsiness [22].For the study 29 subjects are taken and none of them have any physical or mental illnesses. Data is taken from EEG recording signals, and characteristics are compared to determine whether the subject is sleepy or not. In another research EEG-based encode-decoder method is presented for the detection of driver's drowsiness. The working of this model is shown in Fig. 3 [23].

Fig. 3. Working of EEG System for drowsiness detection.

ANN is also used in recent research for drowsiness detection. This research included twenty-one people, of whom ten women and eleven men are included. The scale has a set range of 1 to 15. The range from 0 to 8 indicates that there is no sleep. A notch of 8 to 14 indicates that the individual is showing some signs of sleep, while a score of 15 or above indicates that the person is very sleepy. These participants operate the vehicle for 110 minutes while feeling sleepy. Measurements of functional performance, including emotion rate, eyelid activities, breathing amount, and heavy performance, including rapidity, direction-finding wheel angle location on the way, and time-to-lane adventure, are the foundation of this research. When the driver's condition deteriorates, it forecasts their situation within five minutes [7].

In the latest research [24] fuzzy logic based system is developed to detect the driver's fatigue level on sequence of images and generates alarm. This method used deep learning method for analyzing the image and combined AI and DL for feature extraction. The model gives the classification accuracy of 93.7%. Researchers in [25] use emotion analysis with CNN for accurately detecting driver's drowsiness. This study used two levels CNN for reducing detection time. S. P. Measures [26] in his latest study proposed AI model based MTCNN and GSR for measuring face features and physiological factors. This model use both intrusive and non-intrusive detections. The efficiency of this model was 91%.

III. Research Methodology

The proposed study use CNN Layer model of deep learning for an efficient drowsiness detection. The architecture of CNN model is explained in Fig. 4.

The working of the whole model is illustrated in this section of research.

A. Dataset

The dataset is the important key factor of this research. For the proposed study the dataset is taken from the keggle [27]. This study is using Version No 1 of the dataset uploaded by Serena Raju. The Dataset consists of almost 70 participants (including men and women of different ages) with different driving scenarios. Dataset Contain two folders training and testing and each of the folders have four sub folders (Open Eye, Close Eye, Yawn, and No Yawn). The Details of the dataset is explained in Table I.

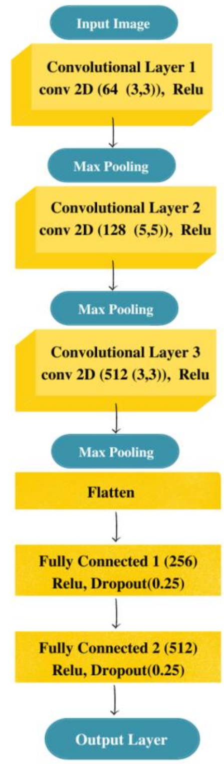

Fig. 4. The layer architecture of CNN models.

TABLE I. Dataset Used for the Proposed Study

Training Dataset (Total Images: 2468 images)		Test Dataset (Total Images: 436 images)	
Close Eye	617 images	Close Eye	109 images
Open eye	617 images	Open eye	109 images
Yawn	617 images	Yawn	109 images
No Yawn	617 images	No Yawn	109 images

The eye data set possibly cover all data possible situations that a driver feels during the driving. The Data set also contains the conditions for a driver who wears glasses. The dataset includes the participants with different feature includes face shape, color, texture, and facial local features.

B. Data Preprocessing

This section of research explains the process of feature extraction and balancing the drowsiness detection dataset [28] [29]. It is the most important part of deep learning algorithms. An efficient and correct dataset is responsible for the efficient results. The dataset used for the study was imbalanced. If the dataset is imbalanced then the classification will not be equally distributed. So the dataset for the proposed study is balanced by Synthetic Minority Over-Sampling Technique

(SMOTE). It is a technique in which the number of minority classes are increased [30][31]. Fig. 5 explains how to create synthetic data points in SMOTE.

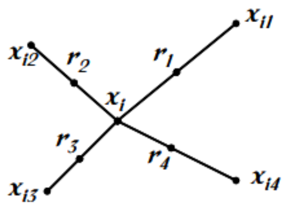

Fig. 5. Creation of Synthetic data points in SMOTE.

The working of SMOTE algorithm is as follows

The algorithm for SMOTE is [32].

- Generate the minority classes of the dataset.
- Generate the oversampling for calculating instances.
- Identify k instance in the minority class and also find its N Neighbor.
- Calculate the distance between these two points N and K
- Multiply the answer with any number exists between 0 and 1 and add this distance in k.
- Repeat the process till required instances.

The benchmark dataset for the purposed study is denoted by D, which is defined as,

$$D=D^{+}\, U D^{-} \tag{2}$$

Here D^{+} considered as drowsy data images while D^{-} is non-drowsy data images and U is the union for both sequences.

C. Feature Extraction

The process of feature extraction includes extracting main face features from the images dataset to process further for drowsiness detection. For the proposed study different feature extraction methods are used for the extracting main features from the dataset. Images plotted for this study were created using the Matplot.lib software. Three sections make up the mat plot lib code. [33]. The photos are plotted on a 2D scale. We scale the photographs to the same size before plotting them to ensure that they line up exactly on the scale. The photos that were utilized for this investigation are 48 × 48 and are presented with a dark backdrop in grayscale. Fig. 6 displays the outcomes of different picture plots made using Matplotlib.

In the proposed study the dataset in consists of 2904 images. The statistical moment used to find the central tendency, probability distribution , dispersion, and symmetry of such dataset [34]. Hahn moment utilize Hahn polynomial for image feature extraction. The mathematical formula for calculating Hahn moment is explaining in Eq. (3).

$$H_{pq} = \sum_{p=0}^{N-1} \sum_{q=1}^{N-1} G'(p,q)\, h_n^{\overline{x,y}}\,(q,N) h_j^{\overline{x,y}}\,(p,N) \tag{3}$$

Raw moment is the statistical moment use to find the position of each image pixel of drowsiness dataset. This is also called crude moment. The raw moment at any random point is calculated by Eq. (4) [35].

$$R_{pq} = \sum_{a=1}^{N} \sum_{b=1}^{N} a^p b^q P'(a,b) \tag{4}$$

In Eq. (4) $P'(a,b)$ the arbitrary point at any two face features (a, b), R_{pq} is the raw moment of these points. Central moment is the Arithmetic mean of the image pixels in the dataset. The arbitrary centroids serves the key feature for finding the probability distribution of the genes [36][37]. The central moment for thee selected dataset is represented by Eq. (5)

$$C_{pq} = \sum_{a=1}^{N} \sum_{b=1}^{N} (a-\bar{x})^p (b-\bar{y})^q P'(a,b) \tag{5}$$

PRIM and RPRIM are the methods for finding the location of image pixels[38]. The position incidence vector deeply describes the combination of pixels in an image. Accumulative Absolute Position Incidence Vector (AAPIV) [39] for finding the nth image pixel in the images of drowsiness detection dataset is determined by Eq. (6) as

$$\beta_N = \sum_{k=1}^{n} \mu_k \tag{6}$$

The reverse AAPIV also work same order as AAPIV work but in the reverse pattern. Different convolutional filters are also applied for extracting the features as explained in Fig. 7.

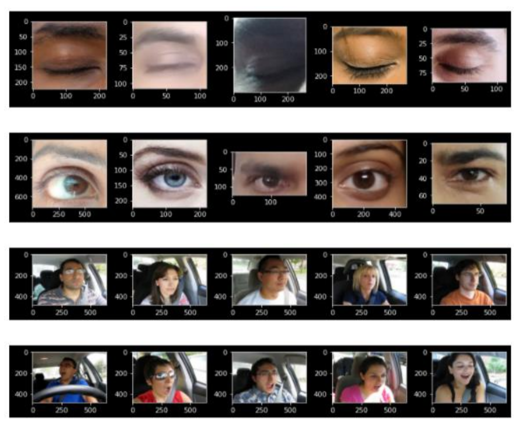

Fig. 6. Results of images plotted using matplotlib.

Operation	Filter	Convolved Image	Operation	Filter	Convolved Image
Identy	$\begin{matrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{matrix}$		Embos	$\begin{matrix} -2 & -1 & 0 \\ -1 & 1 & 1 \\ 0 & 1 & 2 \end{matrix}$	
Sharpen	$\begin{matrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{matrix}$		Sobel	$\begin{matrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{matrix}$	
Outline	$\begin{matrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{matrix}$				

Fig. 7. Different Convolution based 2 dimensional 3×3 filter applied on an images.

D. Tools

For the validation and training of the dataset using Open CV Python, this study uses the Kaggle kernel. The model was developed using Kaggle. The results are then gathered using PyCharm community edition 2020.3.4 and the trained dataset file (Python 3.9). In this research, the face is found using the Haarcascascade file.

IV. Using Working of CNN Model

The working of the model is illustrated in Fig. 8

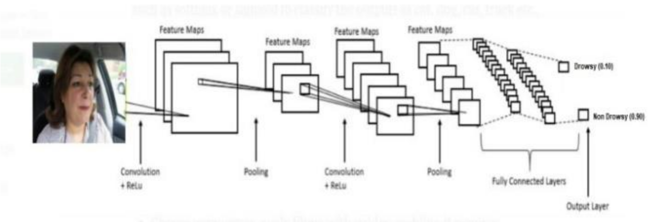

Fig. 8. Working of CNN model.

- Select the images from the drowsiness detection dataset.
- Images were sent to CNN's convolution layer. Different feature extraction techniques and filters are applied on these images.
- Apply the matrix's ReLU activation function to classify the features of the images.
- Max pooling is used to reduce a matrix's dimension size so that the important face features are passed to the next layer.
- Add one more layer of convolutions. (Apply till results are satisfied.)
- Apply the flatten function to the outputs after the necessary outcomes have been achieved in order to transmit them to the completely linked layer.
- Use the softmax activation algorithm to generate the classes and categorize the images (Drowsy or Non-Drowsy)

A basic ConvNet is made up of a number of layers, and each layer performs a differentiable function by converting one volume of activations to another. Each neuron in this coating will be associated to every number in the preceding volume in a conventional neural network. The net output from the forward pass of CNN algorithm is considered by the following equation

$$\text{Net (y)} = \sum_{i=1}^{n} \boldsymbol{x_i w_i} + \text{b} \tag{7}$$

Each layer's g is determined using w(x) +b. If w is the weight vector, b is the bias, and the y is initial function, and the vector **x** is the input. There are several learnable filters in the Conv layers. The driver's image from the dataset passes from each filter so that the system identifies its main features. CNN adds the multiplication values after multiplying a matrix of pixels with a filter matrix, or "kernel." Once all of the pixels have been covered, it goes on to the next set of pixels. After applying the filters on the images ReLU activation function is applied on images. The equation applying the RELU activation function is:

$$RELU = -eV\,max\,(-eV, 0) = 0 \tag{8}$$

$$RELU = +eV\,max\,(+eV, 0) = +eV\,value \tag{9}$$

For each iteration the net output is calculated by Eq. (10)

$$Out\;net\;y = \frac{1}{1+e^{-nety}} \tag{10}$$

Applying RELU to the drowsiness detection dataset make the classes of the images. If the image plot above the graph that shows drowsiness while below the graph shows non-drowsiness. The Max pooling function is used for the suggested research following the activation function. A down sampling technique would be this. The greatest value from the filter is extracted using max pooling. Max pooling is determined by the Eq. (11).

$$Max\;Pooling = \frac{I_x - P}{S} \tag{11}$$

In equation I_x is input x or y shape of the image, p is pooling window size and S represent stride. The data is flattened into a one-dimensional array before moving on to the next layer, which creates a single long featured vector linked to the fully connected layer of the final classification model. Data will enter the fully linked layer's input layer after flattening. The classes are formed from these layers. The CNN Model's output are classes. There are two classifications in the drowsiness detecting system. Drowsy and Non-Drowsy. The softmax activation function determines the likelihood that an input belongs to each class in the dataset. This activation function is used on the CNN fully connected layers for the proposed model. In softmax, the total number of classes equals one. The softmax calculation formula is

$$f\,(y_i) = \frac{e^{yi}}{\sum_k e^{yk}} \quad for\; i = 1\; to\; k \tag{12}$$

The error is determined once one iteration is finished. To assess how effectively the neural network is functioning, error is crucial. The error is minimal if the network is functioning correctly. Error is determined by

Total Error = The Actual output – The Desired output

$$E = Y - \; Y' \tag{13}$$

And the loss function remains calculated through formula

$$L = \frac{1}{2} \sum (Y - Y')^2 \tag{14}$$

Here, **Y** is the productivity that was really received, while **Y**' is the output that was obtained afterward every iteration of the CNN model and each time it was close to the actual output. Following the calculation of the loss, backward propagation is used to update the weights. New weights are computed using backward propagation, and these weights once again flow through the feed-forward neural network to produce the value Y. These iterations continue until the desired output value is obtained. The CNN weights used in backward propagation find out by the equation (15).

$$w_{new} = \mathrm{w}_{\mathrm{old}} - \mu \frac{\Delta \mathrm{L}}{\Delta \mathrm{w}_{\mathrm{old}}} \quad (15)$$

Here, w_{new} is the next updated weight vector, while w_{old} is the previous weight for the updated neuron. Parameter $\boldsymbol{\mu}$ is learning rate and $\frac{\Delta L}{\Delta w_{old}}$ (partial derivative of loss with respect to the old weights). Continuous resampling model based on new weights is used that are calculated by the Eq. (15). And it works until it reaches the global minima [40]. Loss function's derivative and the calculation of old weights is calculated with the chain rule.

$$\frac{\boldsymbol{\Delta L}}{\boldsymbol{\Delta w}} = \frac{\boldsymbol{\Delta L}}{\boldsymbol{\Delta O}} \times \frac{\boldsymbol{\Delta O}}{\boldsymbol{\Delta w}} \quad (16)$$

The acronym for Adam is adaptive moment estimation. It is a technique that uses a stochastic gradient to change the weights and other attributes in neural networks. It is used in the proposed CNN model's back propagation because it is more effective and uses less memory. And it works well in situations where there are lots of data. The optimization method is used in this strategy to produce random variables. Adam employs both the Root Mean Square Propagation (RMSP) and the Adaptive Gradient Algorithm (AdaGrad) properties (RMSP). Adam is a technique for adaptive learning that figures out each person's rate of learning for various parameters. From evaluations of the first and second moments of the gradients, it determines the adaptive learning rates of the individual for various parameters. In both mechanics and mathematics, moments are employed to define the distribution and are described by function.

$$\text{Nth momentum} = \frac{x1s + x2s + x3s \ldots\ldots xns}{n} \quad (17)$$

The function's mean is produced at the first instant when the variable's value is changed from 0 to 1. The variance of the function is similarly explained by the second moment, sometimes referred to as the core moment. The skewness is defined by the third Moment. Adam uses the exponential moving average to estimate the moments. This method is based on the gradient measured on the current mini-beach. The following equations in Adam are used to determine the Momentum and RMSP.

$$\mathrm{w}_{\mathrm{t+1}} = \mathrm{w}_{\mathrm{t}} - \mathrm{m}_{\mathrm{t}} \quad (18)$$

$$\mathrm{m}_{\mathrm{t}} = \beta_1 \mathrm{m}_{\mathrm{t-1}} + (1 - \beta) \frac{\Delta \mathrm{L}}{\Delta \mathrm{w}} \quad (19)$$

m_t, m_{t+1} = aggregate gradient at time t and t+1.

$Wt, Wt+1$,αt,∂L = weights at time t and t+1, learning rate, derivative of Loss

β = Moving average.

$$\mathrm{v}_{\mathrm{t}} = \beta_1 \, \mathrm{v}_{\mathrm{t-1}} + (1 - \beta) \left(\frac{\Delta \mathrm{L}}{\Delta \mathrm{w}}\right)^2 \quad (20)$$

Then the weight update equation for Adam will be

$$\mathrm{w}_{\mathrm{t}} = \mathrm{w}_{\mathrm{t-1}} - \frac{\eta \times \mathrm{m}_{\mathrm{t}}}{\sqrt{\mathrm{v}_{\mathrm{t}} + \varepsilon}} \quad (21)$$

The model is stored in model once the optimizer has been applied. H, which is then utilized to train the algorithm. The verbose is used to print the model's details. Finally, halting conditions are established to protect against model over fitting. The detailed summary of the model is explained in Fig. 9.

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 256, 256, 32)      320
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 128, 128, 32)      0
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 128, 128, 64)      18496
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 64, 64, 64)        0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 64, 64, 128)       73856
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 32, 32, 128)       0
_________________________________________________________________
flatten (Flatten)            (None, 131072)            0
_________________________________________________________________
dense (Dense)                (None, 64)                8388672
_________________________________________________________________
dense_1 (Dense)              (None, 4)                 260
=================================================================
Total params: 8,481,604
Trainable params: 8,481,604
Non-trainable params: 0
_________________________________________________________________
```

Fig. 9. Summary of the proposed model.

V. USING WORKING OF CNN MODEL

The Mathematical model o proposed CNN model is explained as,

$$\mu = -\sum_{i=1}^{h_a} (B_\iota \log(€_c)) \quad (22)$$

where, h_a = classes that depends on application Differentiating Eq. (22) with respect to weights.

$$\frac{\partial \mathrm{a}}{\partial v} \text{ And bias } \frac{\partial \mathrm{x}}{\partial q}$$

Softmax Transformation Function is defined in Eq. (22) as

$$€_c = \frac{\boldsymbol{m^{k_v}}}{\sum_{j=1}^{n} \boldsymbol{\rho^{\kappa_\sigma}}} \quad (23)$$

$$\gamma_l = \sum_{j=1}^{\eta_{out}} (v_{jl} * \varkappa_j)$$

γ_l Is calculated via interrelated weights with the $\varkappa_j$

$$\partial \mathrm{a} = \sum_{j=1}^{\eta_{out}}{}' \sum_{l=1}^{\eta_c} \left(\frac{\partial \mathrm{a}}{\partial \gamma_l} \frac{\partial \gamma_l}{\partial v_{j,l}}\right) \quad (24)$$

$$\frac{\partial €_c}{\partial \boldsymbol{\gamma}_l} = softmax\ derivative$$

$$€_c = \frac{\boldsymbol{m^{k_v}}}{\sum_{k=1}^{\eta_c} \boldsymbol{\rho^{\kappa_\sigma}}}$$

$\boldsymbol{\gamma}_l = \sum_{j=1}^{\eta_{out}} (\omega_{jl} * \varkappa_j)$ Is given as $B_\iota = \gamma_l$

There are two cases, first where I = l, and second I ≠ l, when i=nth unit.

Case1: (i = l)

Quotient rule is applied Eq. (23)

$$\frac{\partial €_c}{\partial \gamma_{(i=l)}} = \frac{\boldsymbol{m^{k_v} \sum_{\kappa=1}^{\eta_c} \rho^{\kappa_\sigma} - m^{k_v} m^{k_v}}}{\boldsymbol{\sum_{\kappa=1}^{\eta_c} \rho^{\kappa_\sigma} * \sum_{\kappa=1}^{\eta_c} \rho^{\kappa_\sigma}}} \quad (25)$$

Taking common $\frac{\boldsymbol{m^{k_v}}}{\boldsymbol{\sum_{\kappa=1}^{\eta_c} e^{\gamma_\kappa}}}$ from Eq. (25), we get

$$\frac{\partial \text{€}_c}{\partial \boldsymbol{\gamma}_l} = \frac{\boldsymbol{m}^{\boldsymbol{k_v}}}{\sum_{\kappa=1}^{\eta_c} \boldsymbol{\rho}^{\boldsymbol{\kappa\sigma}}} \left[\frac{\sum_{\kappa=1}^{\eta_c} \boldsymbol{\rho}^{\boldsymbol{\kappa\sigma}} - e^{\boldsymbol{\gamma}_l}}{\sum_{\kappa=1}^{\eta_c} \boldsymbol{\rho}^{\boldsymbol{\kappa\sigma}}}\right]$$

By taking Anti L.C.M, we acquire

$$\frac{\partial \text{€}_c}{\partial \boldsymbol{\gamma}_l} = \frac{\boldsymbol{m}^{\boldsymbol{k_v}}}{\sum_{\kappa=1}^{\eta_c} \boldsymbol{\rho}^{\boldsymbol{\kappa\sigma}}} \left[1 - \frac{e^{Z\boldsymbol{\gamma}_i}}{\sum_{\kappa=1}^{\eta_c} \boldsymbol{\rho}^{\boldsymbol{\kappa\sigma}}}\right] \qquad \{\because \text{i} = l\}$$

Eq. (22) $\text{€}_c = \frac{\boldsymbol{\rho}^{\boldsymbol{\kappa\sigma}}}{\sum_{j=1}^{n} \boldsymbol{\rho}^{\boldsymbol{\kappa\sigma}}}$ can be modified as.

$$\frac{\partial \text{€}_c}{\partial \boldsymbol{\gamma}_l} = \text{€}_c(1 - \text{€}_c) = \text{€}_c\,(1 - \text{€}_c) \qquad (26)$$

case2 $(\text{i} \neq l)$:

Applying derivative rules for taking the derivative of Eq. (24) w.r.t. γ_l

$$\frac{\partial \text{€}_c}{\partial \gamma_l} = \frac{\frac{\partial}{\partial \gamma_l} \boldsymbol{m}^{\boldsymbol{k_v}} * \sum_{\kappa=1}^{c} \boldsymbol{\rho}^{\boldsymbol{\kappa\sigma}} - \boldsymbol{m}^{\boldsymbol{k_v}} \frac{\partial}{\partial \gamma_l} [\sum_{\kappa=1}^{\eta_c} \boldsymbol{\rho}^{\boldsymbol{\kappa\sigma}}]}{\sum_{\kappa=1}^{\eta_c} e^{\boldsymbol{\gamma\varphi}} * \sum_{\kappa=1}^{\eta_c} e^{\boldsymbol{\gamma\varphi}}}$$

By simplifying,

$$\frac{\partial \text{€}_c}{\partial \gamma_l} = 0 - \frac{\boldsymbol{m}^{\boldsymbol{k_v}} * e^{\boldsymbol{\gamma}_l}}{\sum_{\kappa=1}^{\eta_c} \boldsymbol{\rho}^{\boldsymbol{\kappa\sigma}} * \sum_{\kappa=1}^{\eta_c} \boldsymbol{\rho}^{\boldsymbol{\kappa\sigma}}} = -\frac{\boldsymbol{m}^{\boldsymbol{k_v}}}{\sum_{\kappa=1}^{\eta_c} \boldsymbol{\rho}^{\boldsymbol{\kappa\sigma}}} * \frac{e^{\boldsymbol{\gamma}_l}}{\sum_{\kappa=1}^{\eta_c} \boldsymbol{\rho}^{\boldsymbol{\kappa\sigma}}}$$

As we know that $\text{€}_c = \frac{e^{\boldsymbol{\gamma}_i}}{\sum_{\kappa=1}^{\eta_c} \boldsymbol{\rho}^{\boldsymbol{\kappa\sigma}}}$ and $\text{€}_c = \frac{e^{\boldsymbol{\gamma}_l}}{\sum_{\kappa=1}^{\eta_c} \boldsymbol{\rho}^{\boldsymbol{\kappa\sigma}}}$ so by putting these values in Eq as.

$$\frac{\partial \text{€}_c}{\partial \boldsymbol{\gamma}_l} = -\text{€}_c \lambda_l \quad for(i \neq l) \qquad (27)$$

By summarizing Eq. (26) and Eq. (27)

$$\frac{\partial \text{€}_c}{\partial \boldsymbol{\gamma}_l} = \begin{bmatrix} \text{€}_c(1 - \lambda_i\,)\, for(i = l) \\ -\text{€}_c \lambda_l \qquad for(i \neq l) \end{bmatrix} \qquad (28)$$

$$£ = -\sum_{i=1}^{\eta_c} (B_{\iota} * \log(\text{€}_c))$$

Taking derivative,

$$\frac{\partial \text{a}}{\partial \gamma_l} = -\sum_{i=1}^{\eta_c} \left(Y_\kappa * \frac{\partial}{\partial \gamma_l} log(\lambda_\kappa)\right)$$

$$\frac{\partial \text{a}}{\partial \gamma_l} = -\sum_{i=1}^{\eta_c} Y_\kappa \left(\frac{\partial}{\partial y_k} log(\lambda_\kappa)\right) \frac{\partial \lambda_k}{\partial \gamma_l}$$

$$\frac{\partial \text{a}}{\partial \gamma_l} = -\sum_{i=1}^{\eta_c} \frac{Y_\kappa}{\lambda_\kappa} \frac{\partial \lambda_\kappa}{\partial \gamma_l} \qquad (29)$$

$\frac{\partial \lambda_k}{\partial \gamma_l}$ Is previously measured as the softmax gradient. There are two cases that discussed here $i \neq l$, and $k \neq l$ as in Eq. (27). Now Eq. (28) is distributed into two portions

$$\frac{\partial \text{a}}{\partial \gamma_l} = -\frac{Y_\kappa}{\lambda_\kappa} * \lambda_\kappa\,(1 - \lambda_l\,) - \sum_{\kappa \neq 1}^{\eta_c} \left(-\frac{Y_\kappa}{\lambda_\kappa} * \lambda_\kappa\, \lambda_l\,\right)$$

Where,

$$\sum_{\kappa \neq 1}^{\eta_c} \left(-\frac{\text{Y}_\kappa}{\lambda_\kappa} * \lambda_\kappa\, \lambda_1\,\right) \qquad \text{For} \qquad \kappa \neq 1$$

$$\frac{\text{Y}_\kappa}{\lambda_\kappa} * \lambda_\kappa\,(1 - \lambda_1\,) \quad \text{For} \qquad \kappa = 1$$

We can simplify this,

$$\frac{\partial \text{a}}{\partial \gamma_1} = -\text{Y}_\kappa\,(1 - \lambda_1\,) + \sum_{\kappa \neq 1}^{\eta_c} \text{Y}_\kappa\, \lambda_1$$

We can further simplify this as,

$$\frac{\partial \text{a}}{\partial \gamma_1} = -\text{Y}_\kappa + \text{Y}_\kappa\, \lambda_1 + \sum_{\kappa \neq 1}^{\eta_c} \text{Y}_\kappa\, \lambda_1$$

$$\frac{\partial \text{a}}{\partial \gamma_l} = \lambda_l \left(\lambda_\kappa + \sum_{\kappa \neq 1} Y_\kappa\right) - \lambda_\kappa$$

Where $(\lambda_\kappa + \sum_{\kappa \neq 1} Y_\kappa\,)$ represents 1,

$$\frac{\partial \text{a}}{\partial \gamma_l} = (\lambda_l - Y_\kappa\,)$$

$$\frac{\partial \text{a}}{\partial \gamma_l} = (\lambda_l - Y_l\,)\{\because \kappa = l\}$$

Now put the value of $\frac{\partial £}{\partial \gamma_1}$ in

$$\frac{\partial \text{a}}{\partial v_{j,l}} = \sum_{j=1}^{\eta_{out}} \sum_{l=1}^{\eta_c} \left(\frac{\partial \text{a}}{\partial \gamma_l} \frac{\partial \gamma_l}{\partial v_{j,l}}\right)$$

$$\frac{\partial \text{a}}{\partial v_{j,l}} = \sum_{j=1}^{\eta_{out}} \sum_{l=1}^{\eta_c} (\lambda_l - Y_l\,)\varkappa_j \qquad (30)$$

Where $\frac{\partial \gamma_l}{\partial v_{j,l}} = \varkappa_j$ are representing the input weights. The Differentiation of Loss (£) with respect to weights (ω) for the fully connected layer is formulated in Eq. (30).

VI. Analysis and Discussion

The model succeeds after 20 iterations. The training and test images from each layer of CNN pass from each epoch. The model becomes better with each testing cycle, increasing its ability to compute loss, accuracy, specificity, sensitivity, and Mathew's correlation coefficient. The results of the deep learning algorithms are access with different evaluation measure include accuracy, sensitivity, specificity, loss and MCC values [41] [42]. These are the most important evaluation measure used for binary classification. The mathematical equation for calculating these evaluation methods are explained in Eq. (31) to Eq. (34).

$$Specificity = \frac{TN}{TN+FP} \qquad (31)$$

$$Sensitivity = \frac{TP}{FN+TP} \qquad (32)$$

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN} \qquad (33)$$

$$MCC = \frac{(TP\ X\ TN) - (FP\ X\ FN)}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}} \qquad (34)$$

In the equations:

TN = The number of cases that are correctly identified as drowsy.

TP = The number of cases that are correctly identified as cnon drowsy.

FN = The number of cases that are incorrectly identified as non-drowsy.

FP = The number of cases that are incorrectly identified as drowsy.

Confusion matrix of the algorithm is generated to find the results in the form of accuracy, sensitivity, specificity, and MCC value. Fig. 10 shows the confusion matrix [42] for the proposed results.

The accuracy and loss curve of the results are explained in Fig. 11 and Fig. 12.

Fig. 10. Confusion matrix of the results.

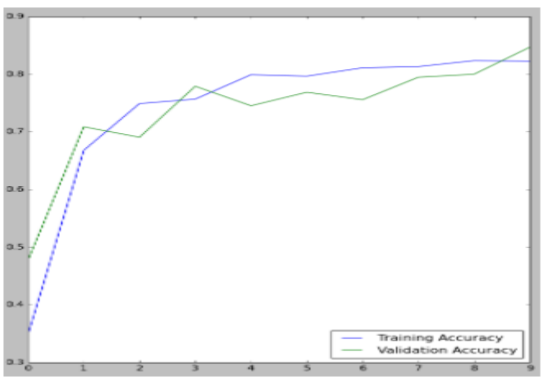

Fig. 11. Accuracy curve.

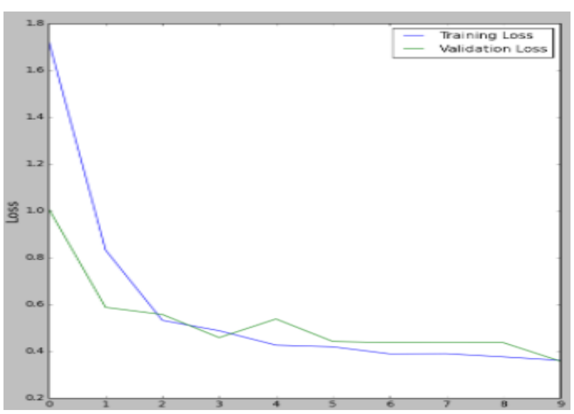

Fig. 12. Loss curve.

It is clearly seen from Fig. 11 that the accuracy of the model is gradually increasing with each iteration.

The effectiveness of the intended system is shown by the graph, which demonstrates that the loss function value for training and testing values is lowering with each iteration while growing accuracy value. In the confusion matrix TP refers to the values that shows the drowsiness and are actual drowsy. TN is the value that is perfectly detected as non-drowsy. FN and FP are the values that are not correctly identified by the model. Table II shows the result of CNN model for the proposed system.

TABLE II. RESULTS OF THE PROPOSED STUDY

Method	Result
Sensitivity	0.94
Specificity	0.95
Accuracy	0.95
MCC	0.82

The proposed study gives the accuracy of 95% for the driver's drowsiness detection. The model is applied on the live camera video to generate the results. The images are taken out of the camera's live video frames and then plotted on grayscale. Face areas are simple to identify in grayscale photos. The regions of the eye are Ex, eye, EW, and eh. Through these areas, the system locates the eye, after which it verifies the closure rate. The technology determines if a user is "Drowsy" or "Attentive" based on the detection of the eye area (Non-Drowsy). The findings that were attained as a consequence of the system's operation are listed below.

Fig. 13. The results of the proposed study on live cam.

These results are obtained by applying the current scenario on different lighting circumstances, driving patters, skin tones and eye conditions with and without using spectacle glasses. Fig. 13 shows the results of the proposed study on live cam.

VII. COMPARISION OF OUR RESULTS WITH STATE OF THE ART TECHNIQUES

A comparison of different methods with their metric, classifier and accuracy from the proposed Method is summarized in Table III.

It is to be seen from the Table III that the maximum accuracy of 93% was obtained by latest studies that used 2D-CNN and EfficientNetB0's architecture for drowsiness detection. Driver's behavior analysis plays an important role in these studies. The proposed study gives the accuracy of 95% with CNN model. In past the driver's drowsiness detection was based on real time detection without training the model on any dataset. Those who use dataset use their own dataset of few images. The proposed study resolves the loophole of the previous study by using an efficient accurate dataset of 2900 images.

TABLE III. COMPARISON OF THE PROPOSED RESEARCH WITH EXISTING TECHNIQUES

Methods	Metrics	Classifiers	Accuracy
Vehicle Based Features	Steering Wheel Movement (SWM)	MANN	88.02%
The Facial Landmarks	Eye Aspect Ratio (EAR) Mouth opening Ratio (MOR)	SVM	92.8%
Physiological and behavioral features	EEG	SVM	80%
Measuring Brain Activity	EEG	Support vector Machine (SVM)	72.7%
Behavioral and Physiological measure	MTCNN with GSR sensor	Hybrid Model of classification	91%
Feature learning	Viola jones method	Convolutional Neural Network(CNN)	81%
Emotion Analysis based CNN Model	2D-CNN	CNN with emotion analysis	93%
Featured learning and Classification (Purposed Model)	Object detection Algorithm(cascade)	Deep Learning(CNN)	90.59%
Deep learning Method	EfficientNetB0's architecture	Combined Deep learning with AI model.	93%
Proposed Study	CNN with viola jones model	CNN Classifier	95%

VIII. CONCLUSION AND FUTURE WORK

The proposed study is going to develop an efficient CNN based system for the continuous detection of driver's drowsiness covering the loophole of the detection time in the previous studies. There were a lot of studies proposed in the past for the detection of driver's drowsiness as explained in Table III of the proposed research. One of the major issues of the previous studies was the delay in detection time and dataset. These studies used physical sensors for monitoring the driver's behavior causes in delay time of detection. The do not detect the face under different circumstances. The accuracy results of detection for these algorithms are not too high.

The main aim of the proposed study is to use a state of the art dataset along with a deep learning algorithm to enhance the efficiency of drowsiness detection. For this, the study uses a big dataset of 2904 images taken from more than 70 participants under all the possible driving scenarios. The dataset is passed from many CNN and statistical filter for an efficient preprocessing. This dataset is used for algorithm training, testing, and validation in order to generate the best results. The proposed study achieves a maximum accuracy of 95%. In past the maximum accuracy of detection was 92.8% that is discussed in Table III. The proposed study gives the highest accuracy of any algorithm for drowsiness detection till date.

The accuracy of the proposed model is the maximum accuracy achieve by any model for drowsiness detection till date. But there exist some loopholes. This model did not detect the drowsiness while driver is wearing shaded glasses. The position of the camera placement over the steering wheel is much important aspect in this scenario. The study covers almost all scenarios of driving but there may exists some scenarios that the study may not cover under detection. This is a weakness for up-and-coming scientists. In the future, a system that uses a dataset larger than the one we now use and can identify tiredness in drivers wearing sunglasses may be developed. By using various deep learning techniques, a system that performs more correctly than the suggested system could exist.

ACKNOWLEDGMENT

The authors extend their appreciation to the Deanship of Scientific Research at the Northern Border University, Arar, KSA for funding this research work through the project number "NBU-FFR-2024-1584-01".

REFERENCES

[1] S. Ilyas, A. A. Shah, and A. Sohail, "Order Management System for Time and Quantity Saving of Recipes Ingredients Using GPS Tracking Systems," IEEE Access, vol. 9, pp. 100490–100497, 2021, doi: 10.1109/ACCESS.2021.3090808.

[2] S. I. and M. K. E. A. Sohail, N. A. Nawaz, A. A. Shah, S. Rasheed, "A Systematic Literature Review on Machine Learning and Deep Learning Methods for Semantic Segmentation," IEEE Access, vol. 10, pp. 134557–134570, 2022, doi: 10.1109/ACCESS.2022.3230983.

[3] Statistics of Road Traffic Accidents in Europe and North America, vol. LVI. 2020. doi: 10.18356/6dd67f42-en.

[4] Z. Zhu, "Real Time Non-intrusive Monitoring and Prediction of Driver Fatigue Manuscript correspondence :," pp. 0–37.

[5] V. a and R. R. Babu, "Facial Emotion Recognition," YMER Digit., vol. 21, no. 05, pp. 1010–1015, 2022, doi: 10.37896/ymer21.05/b5.

[6] Z. H. Zhou, Y. Jiang, Y. Bin Yang, and S. F. Chen, "Lung cancer cell identification based on artificial neural network ensembles," Artif. Intell. Med., vol. 24, no. 1, pp. 25–36, 2002, doi: 10.1016/S0933-3657(01)00094-X.

[7] C. Jacobé de Naurois, C. Bourdin, A. Stratulat, E. Diaz, and J. L. Vercher, "Detection and prediction of driver drowsiness using artificial neural network models," Accid. Anal. Prev., vol. 126, no. July 2017, pp. 95–104, 2019, doi: 10.1016/j.aap.2017.11.038.

[8] M. Ogino and Y. Mitsukura, "Portable drowsiness detection through use of a prefrontal single-channel electroencephalogram," Sensors (Switzerland), vol. 18, no. 12, pp. 1–19, 2018, doi: 10.3390/s18124477.

[9] "Real-Time Eye Blink Detection using Facial Landmarks," Cent. Mach. Perception, Dep. Cybern. Fac. Electr. Eng. Czech Tech. Univ. Prague, pp. 1–8, 2016, doi: 10.1017/CBO9781107415324.004.

[10] K. Dwivedi, K. Biswaranjan, and A. Sethi, "Drowsy driver detection using representation learning," Souvenir 2014 IEEE Int. Adv. Comput. Conf. IACC 2014, pp. 995–999, 2014, doi: 10.1109/IAdCC.2014.6779459.

[11] M. Gulhane and M. P.S, "Intelligent Fatigue Detection and Automatic Vehicle Control System," Int. J. Comput. Sci. Inf. Technol., vol. 6, no. 3, pp. 87–92, 2014, doi: 10.5121/ijcsit.2014.6307.

[12] S. Abtahi, B. Hariri, and S. Shirmohammadi, "Driver drowsiness monitoring based on yawning detection," Conf. Rec. - IEEE Instrum. Meas. Technol. Conf., pp. 1606–1610, 2011, doi: 10.1109/IMTC.2011.5944101.

[13] J. Miao, W. Gao, Y. Chen, and J. Lu, "Gravity-center template based human face feature detection," Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 1948, pp. 207–214, 2000, doi: 10.1007/3-540-40063-x_27.

[14] Y. Qin, B. Tang, and J. Wang, "Higher-density dyadic wavelet transform and its application," Mech. Syst. Signal Process., vol. 24, no. 3, pp. 823–834, 2010, doi: 10.1016/j.ymssp.2009.10.017.

[15] J. Wang et al., “Deep High-Resolution Representation Learning for Visual Recognition,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 43, no. 10, pp. 3349–3364, 2021, doi: 10.1109/TPAMI.2020.2983686.

[16] X. Goldberg, Introduction to semi-supervised learning, vol. 6. 2009. doi: 10.2200/S00196ED1V01Y200906AIM006.

[17] T. Brown, R. Johnson, and G. Milavetz, “Identifying periods of drowsy driving using EEG,” Ann. Adv. Automot. Med., vol. 57, no. June 2018, pp. 99–108, 2013.

[18] University of Waikato, Weka 3 - Data Mining with Open Source Machine Learning Software in Java. 2016. Accessed: Dec. 08, 2020. [Online]. Available: https://www.cs.waikato.ac.nz/ml/weka/%0Ahttp://www.cs.waikato.ac.nz/ml/weka/

[19] M. Maalouf, “Logistic regression in data analysis: An overview,” Int. J. Data Anal. Tech. Strateg., vol. 3, no. 3, pp. 281–299, 2011, doi: 10.1504/IJDATS.2011.041335.

[20] P. Domone, B. Biggs, I. McColl, and B. Moon, “Metals and alloys,” Constr. Mater. Their Nat. Behav. Fourth Ed., no. 69792, pp. 53–81, 2010, doi: 10.4324/9780203927571.

[21] A. A. Shah, F. Alturise, T. Alkhalifah, and Y. D. Khan, “Evaluation of deep learning techniques for identification of sarcoma-causing carcinogenic mutations,” Digit. Heal., vol. 8, 2022, doi: 10.1177/20552076221133703.

[22] T. Tamura and W. Chen, Seamless healthcare monitoring: Advancements in wearable, attachable, and invisible devices, no. January. 2017. doi: 10.1007/978-3-319-69362-0.

[23] S. Arefnezhad et al., “Driver drowsiness estimation using EEG signals with a dynamical encoder–decoder modeling framework,” Sci. Rep., vol. 12, no. 1, pp. 1–18, 2022, doi: 10.1038/s41598-022-05810-x.

[24] E. Magán, M. P. Sesmero, and J. M. Alonso-weber, “applied sciences Driver Drowsiness Detection by Applying Deep Learning Techniques to Sequences of Images,” 2022.

[25] H. Varun Chand and J. Karthikeyan, “Cnn based driver drowsiness detection system using emotion analysis,” Intell. Autom. Soft Comput., vol. 31, no. 2, pp. 717–728, 2022, doi: 10.32604/iasc.2022.020008.

[26] S. P. Measures, “Behavioral and Sensor-Based Physiological Measures,” 2023.

[27] “Face expression recognition dataset | Kaggle.” https://www.kaggle.com/jonathanoheix/face-expression-recognition-dataset (accessed Oct. 29, 2021).

[28] A. A. Shah and Y. D. Khan, “Identification of 4-carboxyglutamate residue sites based on position based statistical feature and multiple classification,” Sci. Rep., vol. 10, no. 1, pp. 2–11, 2020, doi: 10.1038/s41598-020-73107-y.

[29] S. García, S. Ramírez-Gallego, J. Luengo, J. M. Benítez, and F. Herrera, “Big data preprocessing: methods and prospects,” Big Data Anal., vol. 1, no. 1, pp. 1–22, 2016, doi: 10.1186/s41044-016-0014-0.

[30] P. Kaur and A. Gosain, “Comparing the behavior of oversampling and undersampling approach of class imbalance learning by combining class imbalance problem with noise,” Adv. Intell. Syst. Comput., vol. 653, no. January, pp. 23–30, 2018, doi: 10.1007/978-981-10-6602-3_3.

[31] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, “SMOTE: Synthetic minority over-sampling technique,” J. Artif. Intell. Res., vol. 16, no. February 2017, pp. 321–357, 2002, doi: 10.1613/jair.953.

[32] A. Fernández, S. García, F. Herrera, and N. V. Chawla, “SMOTE for Learning from Imbalanced Data: Progress and Challenges, Marking the 15-year Anniversary,” J. Artif. Intell. Res., vol. 61, pp. 863–905, 2018, doi: 10.1613/jair.1.11192.

[33] P. Barrett, J. Hunter, J. T. Miller, J.-C. Hsu, and P. Greenfield, “matplotlib -- A Portable Python Plotting Package,” ASP Conf. Ser., vol. 347, no. June, p. 91, 2005, [Online]. Available: http://adsabs.harvard.edu/abs/2005ASPC..347...91B

[34] S. J. Malebary, R. Khan, and Y. D. Khan, “ProtoPred: Advancing Oncological Research through Identification of Proto-Oncogene Proteins,” IEEE Access, vol. 9, pp. 68788–68797, 2021, doi: 10.1109/ACCESS.2021.3076448.

[35] A. A. Shah, M. K. Ehsan, A. Sohail, and S. Ilyas, “Analysis of Machine Learning techniques for identification of post translation modification in protein sequencing: A Review,” in 4th International Conference on Innovative Computing, ICIC 2021, Lahore: IEEE, 2021, pp. 1–6. doi: 10.1109/ICIC53490.2021.9693020.

[36] W. Zellinger, E. Lughofer, S. Saminger-Platz, T. Grubinger, and T. Natschläger, “Central moment discrepancy (CMD) for domain-invariant representation learning,” 5th Int. Conf. Learn. Represent. ICLR 2017 - Conf. Track Proc., no. Cmd, pp. 1–13, 2017.

[37] X. Shu, Q. Zhang, J. Shi, and Y. Qi, “A comparative study on weighted central moment and its application in 2D shape retrieval,” Inf., vol. 7, no. 1, 2016, doi: 10.3390/info7010010.

[38] A. H. Butt, S. Alkhalaf, S. Iqbal, and Y. D. Khan, “EnhancerP-2L: A Gene regulatory site identification tool for DNA enhancer region using CREs motifs,” bioRxiv, 2020, doi: 10.1101/2020.01.20.912451.

[39] A. H. Butt and Y. D. Khan, “Prediction of S-Sulfenylation Sites Using Statistical Moments Based Features via CHOU’S 5-Step Rule,” Int. J. Pept. Res. Ther., vol. 26, no. 3, pp. 1291–1301, 2020, doi: 10.1007/s10989-019-09931-2.

[40] “Gradient Descent. It is a slippery slope, but promise it… | by Hamza Mahmood | Towards Data Science.” https://towardsdatascience.com/gradient-descent-3a7db7520711 (accessed May 02, 2021).

[41] H. Dalianis, Clinical text mining: Secondary use of electronic patient records. 2018. doi: 10.1007/978-3-319-78503-5.

[42] D. Chicco and G. Jurman, “The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation,” BMC Genomics, vol. 21, no. 1, pp. 1–13, 2020, doi: 10.1186/s12864-019-6413-7.