

# Systematic development of real-time driver drowsiness detection system using deep learning

Tarig Faisal, Isaias Negassi, Ghebrehiwet Goitom, Mohammed Yassin, Anees Bashir,  
Moath Awawdeh

Department of Electrical Engineering, Higher Colleges of Technology, Abu Dhabi, United Arab Emirates

## Article Info

### Article history:

Received May 30, 2021

Revised Dec 14, 2021

Accepted Dec 30, 2021

### Keywords:

Classification

Convolutional neural network

Driver drowsiness

Haar cascade

Hyperparameters optimization

## ABSTRACT

Advancements in globalization have significantly seen a rise in road travel. This has also led to increased car accidents and fatalities, which become a global cause of concern. Driver's behavior, including drowsiness, contributes to many of the road deaths. The main objective of this study is to develop a system to diminish mishaps caused by the driver's drowsiness. Recently deep convolutional neural networks have been used in multiple applications, including identifying and anticipate driver drowsiness. However, limited studies investigated the systematic optimization of convolutional neural networks (CNNs) hyperparameters, which could lead to better anticipation of driver drowsiness. To bridge this gap, a holistic approach based on the deep learning method is proposed in this paper to anticipate the drivers' drowsiness and provide an alerting mechanism to prevent drowsiness related accidents. To ensure optimal performance achievement by the system, a database of real-time images preprocessed via Haar cascade's classifiers is used to systematically optimize the CNN model's hyperparameters. Different metrics, including accuracy, precision, recall, F1-score, and confusion matrix, are used to evaluate the performance of the model. The training evaluation results of the optimal model achieved an accuracy of 99.87%, while the testing results accurately classify the drowsy driver with 97.98%.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



## Corresponding Author:

Tarig Faisal

Department of Electrical Engineering, Higher Colleges of Technology

Ruwais, Al Dhafrah, Abu Dhabi, United Arab Emirates

Email: tfaisal@hct.ac.ae

## 1. INTRODUCTION

With advancements in world development, vehicular movement is becoming prominent, contributing significantly to traffic accidents, traffic injuries, and loss of lives. Various human factors contribute to these accidents, including the recklessness and fatigue of drivers. Drowsiness related accidents have all the earmarks of being more dangerous because of the higher speeds involved in distraction and the driver being unable to take any counteraction. The World Health Organization reported that more than 1.35 million are killed every year, and up to 50 million are injured in road traffic crashes [1]. It's reported by the National Highway Traffic Safety Administration (NHTSA) that drowsy driving contributed to 2.3% to 2.5% of all fatal crashes [2]. The development of a system for the recognition or identification of driver drowsiness is crucial for accident prevention and road safety at large. In general, the loss of awareness due to tiredness causes a few changes in the human body and activities. These side effects motivated many researchers to develop strategies to measure the drowsiness level and monitoring of the behavior of drivers [3]–[5].

Object detection is one of the most common tasks in computer vision and states to determine the existence or absence of prominent features in image data. Once the features are detected, an object is

classified as belonging to one of a predefined set of classes. This later operation is known as object classification. Object detection and object classification are fundamental building blocks of artificial intelligence. The introduction of the deep learning technique has created a significant impact on object detection and classification due to its comparatively high accuracy and speed [6]–[8]. Convolutional neural networks (CNN) is one type of deep learning (DL) model for the processing of data like the detection and classification that has a grid pattern. Inspired by the animal visual cortex organization [9], [10], CNN is designed to learn features from low- to high-level patterns. Deep convolutional neural networks have been used recently in multiple applications for identifying and anticipate driver drowsiness [11]–[13]. In order to detect the driver's drowsiness, a deep neural network-based model is proposed in [13]. The driver's face was extracted from the video and passed as input to the model containing three stages of CNN, a convolutional control gate-based recurrent neural network (ConvCGRNN), and a voting layer. In the proposed system, the CNN layers interpret various facial expressions based on the global data sets while the ConvCGRNN determines the temporal dependencies, and finally, the voting layer determines the level of drowsiness. Experimental analysis based on real-time environment showed an accuracy of almost 85%. For identifying the sleepiness of drivers, the CNN based recurrent neural network (RNN) and long short-term memory (LSTM) were found to be very successful [12]. In this study, the information was extracted from the input images through the CNN and the features were interpreted across consecutive frames via the LSTM. The system achieved an accuracy of 88.5%. In [11], object recognition was used along with deep learning in order to detect the distraction and drowsiness of drivers in real time. Ear aspect ratio (EAR) and mouth aspect ratio (MAR) were computed using 68 facial landmarks to identify the driver's sleep probability. Object detection technique, namely, single shot multi-box detector (SSD) based on convolution neural networks, was used to determine the objects or environmental causes that may cause a distraction to the driver. The system offered an accuracy of 90% under experimental conditions.

Viola-Jones object detection framework proposed in 2001 [14] is a machine-learning object detection technique that has been used successfully to identify objects in an image or video. In drowsiness detection, the Viola-Jones object detection framework was implemented in many applications [15], [16]. A real-time system to detect drowsiness using Android mobile application was discussed in [15]. The study used Haar feature-based cascades for extracting landmark coordination from images and transfer them to the multilayer perceptron classifiers to identify drowsy drivers. 80% accuracy was achieved by the proposed model. One of the aspects of monitoring the behavior of the driver for drowsiness or sleepiness is yawning detection. Using Viola-Jones computational theory, a smart camera with computer vision was utilized to detect yawning by detecting the changes in the driver's mouth [16]. The model was tested on a real platform named CogniVue APEX, which shows improvements in reliability and accuracy when compared to existing methods in the detection of yawning. In this study, the camera for yawning detection was placed on the front mirror or the dashboard, and its output was given to the embedded platform, which displays the results on the monitor. The combination of CNN and Viola-Jones object detection Framework has also been in some applications for drowsiness detection [17], [18]. Viola-Jones face and eye detection algorithm along with deep CNN were used to develop a system for the identification of driver drowsiness [18]. 96.42% accuracy achieved by the proposed system utilizing 2,850 images for training, validation, and testing data. A similar technique with more features was used by [17]. The system achieved an accuracy of 88% for subjects with glasses and more than 85% for category night without glasses. The study indicated that on average more than 83% accuracy was achieved in all categories.

Even though some of those studies have achieved good results for identifying and anticipate driver drowsiness, however, limited studies have investigated in-depth systematic optimization of the CNN hyperparameters, which could lead to higher drowsiness identification performance as well as practical implementation. Accordingly, the main objective of this study is to develop an effective driver drowsiness identification system considering the optimization of the CNN hyperparameters as well as the incorporation of the Viola-Jones object detection framework. The developed system will assist in diminishing avoidable mishaps caused by the driver's drowsiness and save many lives, and this has served as one of our motivations to propose this work. The paper has four sections. In the first section, works related to driver drowsiness detection techniques were reviewed. The methodology in the second section discusses the proposed system, implemented techniques, database, and system evaluation methods. The third section presents the results and is followed by the discussion and conclusion.

## 2. METHODOLOGY

### 2.1. Proposed system

In this research, the primary aim is to develop a Real-Time driver drowsiness detecting system based CNN system to identify whether the driver is drowsy or not. The proposed system flowchart is shown in Figure 1. As shown in the figure, the system starts by receiving inputs from a camera mounted in front of

the driver. The camera continuously feeds a stream of frames of the face of the driver. Then the face feature of the driver is detected, and the location of the eyes is identified. The eye's location is then extracted from the face and preprocessed. The processed image is fed to a trained CNN model to classify it into two categories: "open eyes state" and "closed eyes state." Based on predefined criteria, the system determines whether the output obtained from the CNN model belongs to a drowsy driver or not. Finally, a prototype alert system/mechanism is used to notify the driver in case the driver is found drowsy.

Figure 2 shows the process followed for identifying the optimal CNN model. The first phase starts with the image extraction and preprocessing, which was accomplished using the Viola-Jones object detection framework. The second stage is identifying the optimal CNN hyperparameters, leading to determining the optimal model that accomplishes the best performance. In this stage, four hyperparameters, namely kernel size, Maximum pooling size, learning rate, and a number of epochs, were investigated. Finally, the system was tested experimentally using a real-time date.

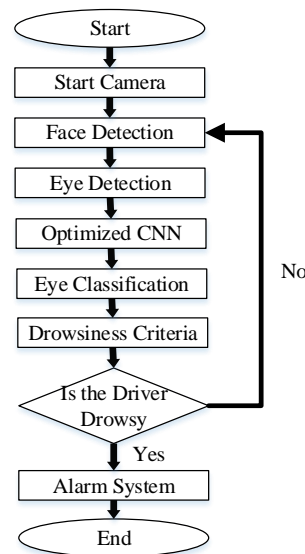


Figure 1. Overall flowchart of the system

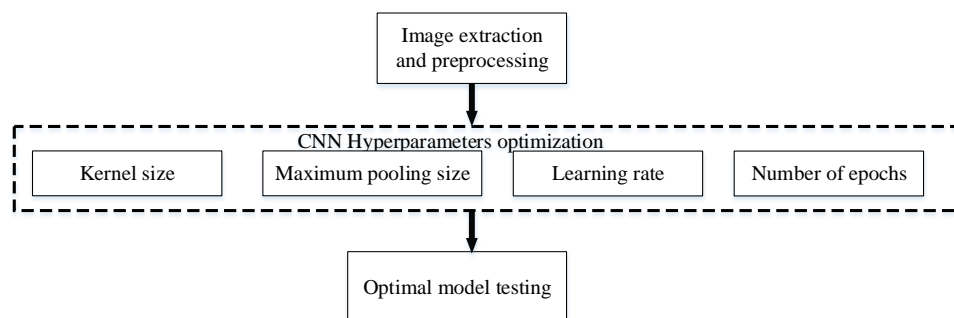


Figure 2. CNN hyperparameters optimization

## 2.2. Image extraction, preprocessing, and database

The flowchart for the procedures of collecting and preprocessing the database is shown in Figure 3. In preparing our data set, we used 20 volunteers from those 7 of them were with eyeglasses. Initially, two types of videos were recorded for each person; one when they are awake and the other when they are feeling drowsy. From the recorded videos of each person, eye images were detected and extracted using Viola-Jones object detection Framework and our advanced formula, which will be discussed later. In total, 4,000 images per person were collected via the camera. Each extracted image has a size of 300×300 pixels. In addition, 20,000 more foreign images are added from the internet with good resolution [19], to obtain better accuracy.

In order for the image to fit the proposed model, the size of the image should be reduced. Therefore, the Numpy library was used to convert the images to an array with a dimension of 100×100 pixels. Then the dataset was normalized to expedite the training. The normalization of the dataset was performed by dividing the RGB image pixel by 255 to get a grayscale image that simplifies the algorithm and reduces the computational requirements. The OPEN CV library with python was used to implement the Viola-Jones object detection framework with the Haar cascade algorithm.

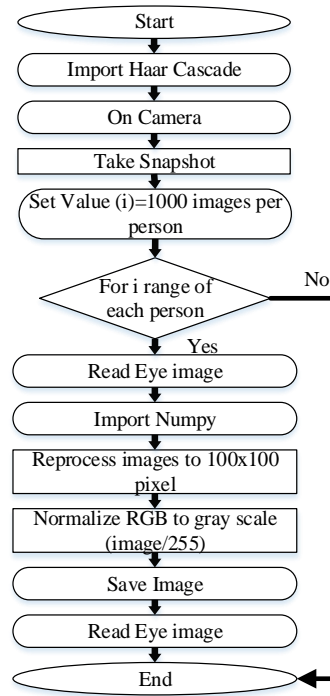


Figure 3. Dataset flowchart

The haar cascade algorithm was used for face detection. Then the eyes, which are the region of interest (ROI) was estimated and extracted since we are interested only in the eye portion of the image rather than the whole face because they are more informative. Haar algorithm provides four parameters to identify the position of the faces namely  $x_0$ ,  $y_0$ ,  $W$  and  $H$ ,  $x_0$ , and  $y_0$  are the starting point for the used rectangle to identify the face.  $W$  is the width of the face rectangle, formed by the horizontal distance from the point  $(x_0, y_0)$  to the right.  $H$  is the height of the face rectangle, which formed by the vertical distance from the point  $(x_0, y_0)$  downward direction. Figure 4 displays those parameters.

As shown in the Figure 4, to estimate the eyes region, the following positions need to be identified:  $(x_1, y_1)$  which is the starting point for the eye rectangle, the second vertex point  $(x_2, y_1)$  found to the right of the starting point, and the third vertex point  $(x_1, y_2)$  found below the starting point. Those positions of the three vertices are enough to draw a rectangle around the eyes. In this study, (1)-(4) is used to identify the eye's rectangular vertices based on the four parameters obtained from the Haar cascade algorithm. The coefficients shown in the above equations were determined empirically in this study.

$$x_1 = x_0 + 0.084W \quad (1)$$

$$x_2 = x_0 + 0.916W \quad (2)$$

$$y_1 = y_0 + 0.25H \quad (3)$$

$$y_2 = y_0 + 0.5H \quad (4)$$

It is worth mentioning that the estimated and extracted eyes region size and position change proportionally with the identified face since the size of the detected face changes with the distance of the person in front of the camera (the identified face rectangle enlarges if a person is close to the camera and vice versa).

versa). Finally, only the identified eye ROI was extracted from the face and fed to the CNN for training and testing. Figure 5 shows samples from the database of extracted awake and sleep images. Accordingly, around 100,000 images were collected. From these, 62,000 images were without eyeglasses, and the rest 38000 images were collected with eyeglasses to ensure the system's efficiency. 80% of the data were used for the training process and 10% for the validating; the remaining 10% was used for the testing dataset.

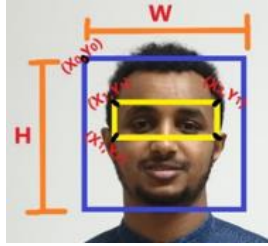


Figure 4. Sample photo showing parameters and points used for faces and eyes



Figure 5. Randomly taken samples from the dataset of wake and sleep

### 2.3. Convolutional neural network structure

Convolutional layers, pooling layers, and fully connected layers are the main types of layers included in the CNN. It primarily starts with the convolutional layer, which is the most important layer in CNN. In this layer, the basic different visual features are extracting from the input image. Two different operations are performed in the Convolutional layer; linear operations are represented by the convolution operation, and the activation function reflects the nonlinear operations. The convolution operation aims to extract different input image features by kernel convolving (sliding) throughout the whole input image.

The convolution process begins from the left-hand corner top portion of the input image and gradually and sequentially moves to the top right and then moves downwards in the same manner, covering the entire image. The stride determines the number of elements that would be convolved by the kernel. Between the tensor (which refers to the portion of the input image) and the kernel, an element-wise multiplication and accumulation operation would occur at every stride. The feature map,  $Y(i, j)$  would be produced from the output values of this operation and is defined by the following equation,

$$Y(i, j) = (I * F) [i, j] = \sum_a \sum_b I[a, b] F [i - a, j - b] \quad (5)$$

where  $F$  represents the kernel,  $I$  represents the input image,  $j$  is the convolved matrix's index columns, and  $i$  is the convolved matrix's index rows.

The extracted features are determined by the number of kernels; 1 mapped feature would be produced from 1 kernel, whereas  $T$  mapped features would be produced from  $T$  kernels. Hence the input image, kernel size, and stride size would determine the size of the feature map. If the input image  $I$  has a size of  $m \times n$ , the Kernel (filter)  $F$  has a size of  $u \times v$ , and the number of strides is  $g$ , the  $Y$  feature map size is calculated as:

$$Y(i, j) = Y \left( \frac{m-u}{g} + 1, \frac{n-v}{g} + 1 \right) \quad (6)$$

The activation function is the nonlinear operation of the CNN, which represents the second stage of the convolution operation. It is used to improve the learning capability of the neural network for any complex data based on the input. Although many activation functions exist, the rectified linear unit (ReLU) is a very popular type used as it is speedy and efficient [20]. It gives a null value when the input value is negative, and it is represented by the (7):

$$H(z) = \begin{cases} 0 & \text{if } z < 0 \\ z & \text{if } z \geq 0 \end{cases} \quad (7)$$

where  $z$  is the input of the function.

The pooling layer is located between the convolution layers. It is used to preserve the significant information that minimizes the computation time for the following layer and reducing the features map.

Likewise, the overfitting of the network is also distorted when the learnable parameters are reduced [21], [22]. Several pooling techniques have been used, including maximum pooling, average pooling, and sum pooling. To discard the insignificant features and acquire the maximum value within the pooling window, the maximum pooling is used in this study.

The above convolutional and pooling process is shown in Figure 6, which is the schematic representation of the network architecture used in our proposed system to identify the drowsiness drivers. The network comprises three convolutional blocks (gray) and three max-pooling (orange) layers. An input image of size 100×100 pixels is convolved with thirty-two 10×10 filters (with unitary strides) to produce thirty-two 100×100 feature maps (C1 in gray). These feature maps are then passed to 15×15 max-pooling operations that downsize the feature maps to thirty-two 64×64 feature maps (P1 in orange). Next, these feature maps are passed through the second convolution layer of sixty-four 10×10 filters (with unitary strides), yielding sixty-four 32×32 features maps (C2 in gray). The second 15×15 max-pooling operations downsize the produced feature maps to sixty-four 22×22 feature maps (P2 in orange). The last convolution operation resulted in two hundred fifty-six 18×18 feature maps (C3 in gray), which is then downsized to two hundred fifty-six 9×9 feature maps (P3 in orange) using the last pooling operation. The nonlinear operation in all convolution blocks is performed using the ReLU activation functions. The third and final layer of the CNN is the fully connected or dense layer. It receives the input volume of information from the output of the preceding layer and generates a Q-dimensional vector output by calculating the probability score. Q represents the number of output classes, which is 2 classes for our study. This process is accomplished by flattening the input volume received from the preceding layer to a vector which is then fed to single or multiple fully-connected layers. The fully connected layers comprise multiple neurons fully connected to each and every input and output through learnable weight. The fully-connected layers comprise multiple neurons entirely connected to all inputs and outputs via learnable weight. The fully-connected layers' output is enjoined to an activation function that yields Q-dimensional vector probability scores output equivalent to the classification target by normalizing the outputs. One of the most commonly used activation functions is the Softmax, and its output estimation is given in the (8).

$$\phi(x)_i = \frac{e^{x_i}}{\sum_{k=1}^K e^{x_k}} \quad \text{for } i = 1, \dots, K \quad (8)$$

where  $\phi()$  represents Softmax activation function,  $x$  is the input and  $K$  in the number of the classes.

The above process is presented in the seventh layer in the proposed system, as shown in Figure 6. The feature maps produced from the sixth layer are flattened passed through one fully connected layer with 512 nodes. Finally, the states of the eye open or close take place in the output layer via a softmax activation function. Table 1 summarize CNN layers parameters.

CNN training aims to reduce the variance between the desired and the generated outputs by obtaining suitable learnable parameters values, including kernels, weights, and biases. Calculating these variances can be accomplished by many types of loss functions; however, mean square error (MSE) loss functions are used in our application, as it is the most commonly used. The MSE can be calculated by (9):

$$\psi = \frac{1}{r} \sum_{n=1}^r (O_n - \hat{O}_n)^2 \quad (9)$$

where  $r$  is the number of the samples,  $\hat{O}_n$  is the desired output and  $O_n$  is the calculated output.

The training is commonly performed using the stochastic gradient descent (SGD) and the stander backpropagation [23]. The SGD is a commonly used optimization algorithm since other algorithms is computationally expensive [19], [21]. The training starts by forward propagation, of which the MSE is calculated and followed by the back propagation, where the learnable parameters are updated. It worth mentioning that the selected training algorithm determines the ways to update the learnable parameters. The iteration is increased by one after all the parameters are updated. The preceding steps are repeated until a given number of iterations are completed or a satisfactorily small MSE value is attained. The SGD is updating the learnable parameters to reduce the loss function using the following formula:

$$P \leftarrow P - \xi \frac{\partial \psi}{\partial P} \quad (10)$$

where  $P$  is the learnable parameters, and  $\xi$  is the learning rate.

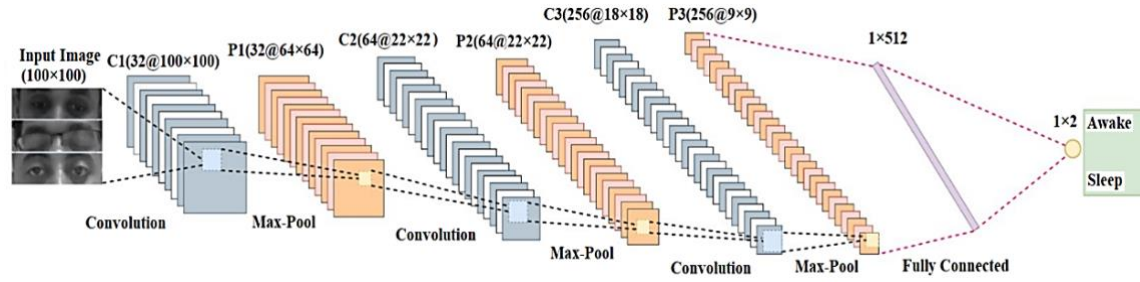


Figure 6. CNN architecture of the system

Table 1. Description of the CNN parameters

Layer	Size	Other parameters
Input	100×100×1	Grayscale image=100,000 images
Convolution C1	100×100×32	Window size=15
Max Pooling P1	64×64×32	Window size=10
Convolution C2	48×48×64	Window size=15
Max Pooling P2	22×22×64	Window size=10
Convolution C3	18×18×256	Window size=15
Max Pooling P3	9×9×256	Window size=10
Fully Connected	1×512	

## 2.4. CNN hyperparameters optimization

Since the tunable hyperparameters are not updated during the training process, the CNN model optimization was performed by obtaining the optimal hyperparameters, which could lead to effective performance. The following hyperparameters were examined in our study: kernel size (convolutional window size), maximum pooling size, learning rate, and the number of epochs. Hence, four steps were iterated to attain the optimal model by fine-tune the hyperparameters. At every step, we varied the hyperparameters that needed to be optimized and fixed the other parameters. The optimal parameters selection criteria were based on the highest training and validation accuracy and the minimum MSE square error.

In the testing phase, the confusion matrix was used to assess the model performance by summarizing the optimal model predictions. The use of the confusion matrix reduces any prejudices that may arise due to unbalanced data [24]. Based on the confusion matrix, the precision, accuracy, recall, and F1 score were calculated. Those parameters have been used in many applications to calculate the effectiveness for the classification of the CNN [25]–[27]. The accuracy calculates the percentage of images correctly predicted by the model, which can be calculated as:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \times 100\% \quad (11)$$

where: TN is True negatives, TP is True positives, FN is False negatives, and FP is False positives.

In our model, the open eyes were labeled as positive, while the closed eyes were labeled as negative. TP means that the model correctly predicted the open eyes as open eyes. In contrast, FP indicates that the model predicted the closed eyes as the open eyes, and the same thing applies to TN and FN.

The correctness of the classification was calculated using the precision (12):

$$Precision = \frac{TP}{TP + FP} \times 100\% \quad (12)$$

the effectiveness for the classification was calculated using the recall (13):

$$Recall = \frac{TP}{TP + FN} \times 100\% \quad (13)$$

finally, the F1 score was calculated (14):

$$F1 \text{ score} = 2 \times \frac{Precision \times Recall}{Precision + Recall} \times 100\% \quad (14)$$

## 2.5. Drowsiness identification criteria

In the development of the proposed system, it's very crucial to differentiate whether the eyes are blinking or totally closed. Generally, a single blink can be divided into four phases: closing, closed, early opening, and late opening [28]. Calculating the duration of the human blink is challenging, mainly calculating the duration of the eye closing during blinking. Many experiments were conducted in our study to calculate the duration of the eyes closing during blinking. In those experiments, the eyes were considered as a closed eye state, if it's fully closed or more than 90% closed. Based on this assumption, 0.2 seconds were found as an average duration a person can close eyes during blinking. In the experiment, the Logitech webcam camera was used, which was capable of taking 30 frames per second. Accordingly, based on the estimated eyes blinking duration, the camera can take up to 6 frames in 0.2 seconds. As a result of this calculation, it can be estimated that a person can be considered asleep if the eye closes for seven consecutive frames (0.234 seconds) or more (one frame was added for confirmation). Therefore, the drowsy criteria were set in the proposed system in a way that if seven consecutive frames are classified by the CNN model as "closed eyes state," it reflects the drowsiness of the driver.

## 3. RESULTS

### 3.1. Hyperparameters optimization

The optimization results of the CNN Hyperparameters are shown in Figures 7 to 10. Figure 7 shows the results of determining the optimal number of epochs. As shown in the figure, the initial training accuracy was 50.42%, and validation accuracy was 45%. The initial loss of the training and validation was 10.985 and 11.985, respectively. From Figure 7, it can be observed that the training and validation accuracy is increased, and the loss is decreased when the number of epochs is increased. The highest training and validation accuracy with the lowest loss was achieved with the number of epochs of 100. The attained training accuracy was 70.219%, while the validation accuracy was 65.12%. Therefore, the model with 100 epochs was chosen.

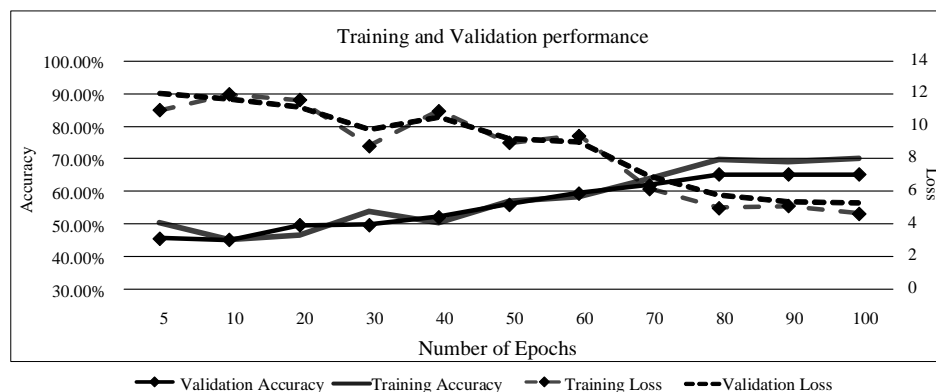


Figure 7. Training and validation performance with varying the number of epochs

Figure 8 shows the effects of varying the learning rate. The learning rate value was varied from 0.0000001 to 1. Figure 8 depicts that the learning rate of 0.01 produced the highest training accuracy of 84.348% and validation accuracy of 82.106%. Additionally, with the same learning rate, the network produced the lowest training and validation loss of 3.04. Accordingly, 0.01 was chosen as the optimal learning rate.

Determining the optimal convolutional window size is shown in Figure 9. The figure shows that convolutional window size of 15 produced training accuracy of 84.723%, validation accuracy of 86.9%, training loss of 2.813, and validation loss 2.9. Increasing the window size for more than 15 doesn't improve the network accuracy; therefore, it was chosen as an optimal value.

Figure 10 shows network performance for various max-pooling window size. The value of the max-pooling window was varied from 3 to 12. As shown in Figure 10, with a max-pooling window size of 10, the highest training and validation accuracy of 99.87% and 99.63% were achieved, respectively. On the other hand, the training loss is 0.871, while the validation loss is 0.991.

In light of the above results, the hyperparameters optimization effectively increased the training and validation accuracy from 50.42% and 45% to 99.87% and 99.63%, respectively. Accordingly, the optimal



hyperparameters were chosen as 100, 0.01, 15, and 10 for the number of epochs, learning rate, Convolutional window size, and max-pooling window size, respectively.

### 3.2. Model Performance evaluation

Tables 2 and 3, present the confusion matrix and the testing performance results obtained by using the optimal network and 10000 unused images in the testing phase. Those images are comprised of 5000 open eyes images and 5000 closed eyes images of different individuals in the data set. In the confusion matrix, each row represents the actual class, and each column represents the predicted class. As shown in the tables, the model perfectly classified the input data with 97.98% accuracy, 98.06% precision, 97.903% recall, and 97.981% F1 Score.

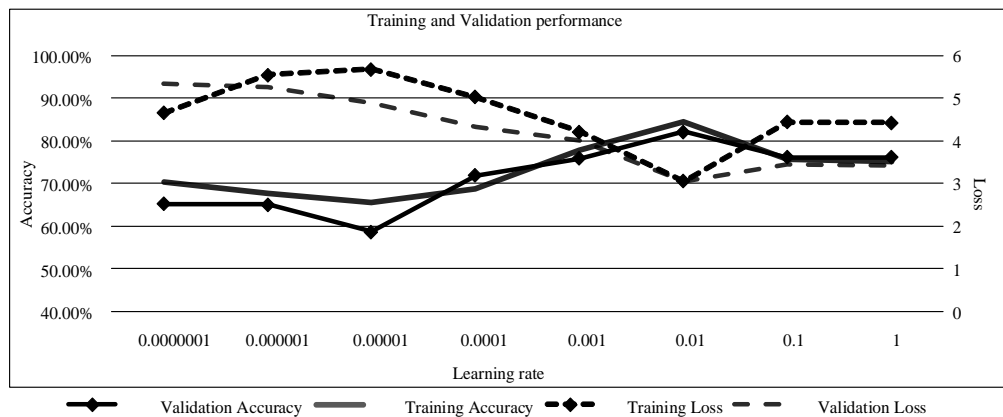


Figure 8. Training and validation performance with varying the learning rate

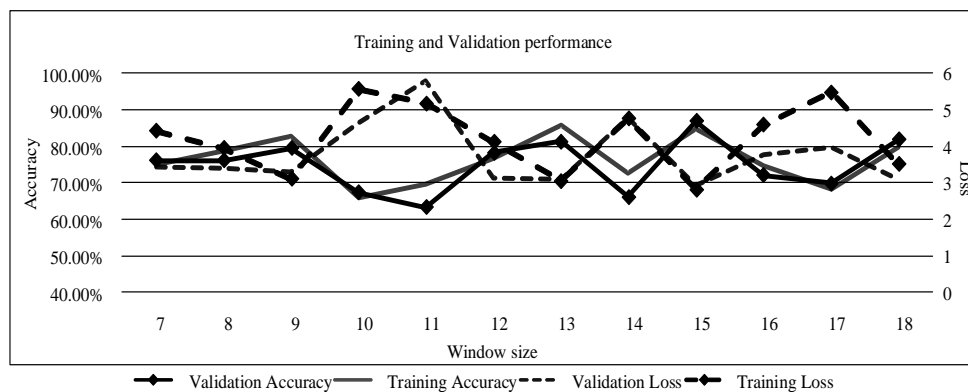


Figure 9. Training and validation performance with varying the convolutional window size

### 3.3. Prototype development

The major components used in the developed prototype system include Logitech webcam, NVidia Jetson Nano, LCD monitor, and Alert system (alarm speaker and Car Seat Vibrator). NVidia Jetson nano microprocessor is one of the powerful microprocessors available in the market which are capable of processing real time video for artificial intelligence systems. It has a 128 core NVidia Maxwell processor with a speed of 1.43 GHz. It has 4 GB 64-bit low-power double data rate (LPDDR4) memory capacity. The interface of the Nvidia Jetson Nano started by initially preparing a 64 GB microSD card to serve as a boot device and main storage. Then the image that contains bootable files and an ubuntu operating system are copied into the SD card using launcher Etcher before inserting the memory card into Nvidia Jetson Nano. Subsequently, the peripherals connected to the Nvidia Jetson Nano and the system was booted the model was uploaded our model.

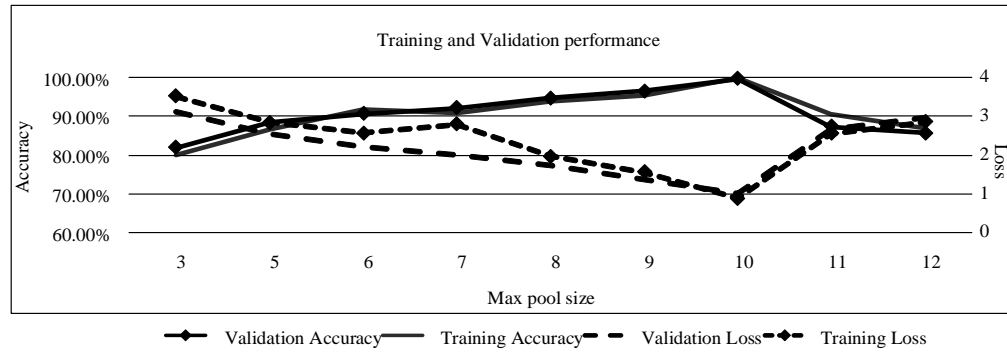


Figure 10. Training and validation performance with varying the Max\_pool window size

Table 2. Confusion matrix of the proposed model

		Predicted Class	
		Positive	Negative
Actual Class	Awake	4903	105
	Drowsy	97	4895

Table 3. Testing performance

Parameters	Values
Number of Images	10000
Accuracy(%)	97.98%
Precision(%)	98.06%
Recall(%)	97.903%
$F_1$ Score(100%)	97.981%

Accordingly, the developed system operates: Firstly, the system is powered, and it takes three seconds to boot. A mounted camera starts to continuously record video and monitors the person (driver) present in front of it. The captured raw video is then sent to the NVidia Jatson nano for feature extracting and processing.

The processing starts by extracting frames from the series of frames included in the captured video and then reshape and resize them. The processed image is then fed to the developed CNN model to identify the state of the eyes, whether they are closed or opened. If the eyes are identified as open, the above process restarted again. However, if the state of the eye is closed, the consecutive frame is used to differentiate whether the driver is blinking or drowsy. This process is continued until the number of consecutive frames is equal to the predefined ones. Once confirming the eyes are closed, the alert system composes a seat with a motor (to simulated the car seat vibrator) and an alarm speaker is activated, and a warning sign is displayed on the LCD monitor. The alert system is interfaced with NVidia via relays. The alert system is deactivated if the eyes are opened or the engine is turned off.

#### 4. DISCUSSION

Table 4 compare the proposed system with other latest developed systems. We have considered the latest studies that have similar feature to our system to identify the gaps and optimal solutions. Some of those systems used available online database where other system including our system, developed their own dataset. Studies used the online dataset used subjects with and without eyeglasses, while the other studies used their own developed dataset used only subjects without eyeglasses. However, in our studies, we used subjects with and without eyeglasses even though we developed our dataset. It can also be noticed that most of those studies didn't present in-depth any optimization for the CNN Hyperparameters or the experimental identification haar cascade classifier parameters. They rather focus on the application of the method. On the other hand, our study presented systematic optimization of the CNN Hyperparameters and the empirical determination of the haar cascade algorithm parameters. As a result, as shown in Table 4 our system outperformed other system's accuracy. On the other hand, one of the limitations of our system compared with other systems, especially those studies that used the available online database, is that our study was based on daytime lighting since we did not use a night vision camera.

Table 4. Brief comparison of the existing and the proposed system

Reference	Implemented system	Feature	Performance
R. Jabbar <i>et al.</i> , 2018 [15]	<ul style="list-style-type: none"> <li>Driver sleepiness identification method using CNNs to extract information from images and long short term memory networks (LSTM).</li> <li>Data for subjects with and without eyeglasses was used</li> </ul>	<ul style="list-style-type: none"> <li>Dlib library is used to extract landmark coordination from images</li> <li>Android mobile application, Java Native Interface (JNI)</li> <li>Available online National Tsing Hua University (NTHU) Driver Drowsiness Detection Dataset was used</li> </ul>	<ul style="list-style-type: none"> <li>80.9% accuracy was achieved</li> </ul>
H. Vanjani <i>et al.</i> , 2019 [12]	<ul style="list-style-type: none"> <li>Deep neural network (DNN) for detecting driver drowsiness in videos.</li> <li>Subjects included with and without eyeglasses</li> </ul>	<ul style="list-style-type: none"> <li>Stander Viola-Jones Haar-Feature was used for face detection.</li> <li>Adam Optimizer was used the CNN model for feature extraction and LSTM for interpreting the features across consecutive frames.</li> </ul>	<ul style="list-style-type: none"> <li>88.5% accuracy was achieved</li> </ul>
T. VU <i>et al.</i> , 2019 [13]	<ul style="list-style-type: none"> <li>Driver drowsiness detection based on eye state.</li> <li>Viola-Jones face detection algorithm with CNN classifier.</li> <li>Subjects included without eyeglasses</li> </ul>	<ul style="list-style-type: none"> <li>MTCNN for face detection, and a correlation function in dlib for tracking</li> <li>CNN, a ConvCGRNN, and a voting layer are used to developed the system.</li> </ul>	<ul style="list-style-type: none"> <li>84.81% accuracy was achieved</li> </ul>
V. Chirra <i>et al.</i> , 2019 [18]	<ul style="list-style-type: none"> <li>Anticipation of driver drowsiness by applying a recurrent neural network over a sequence frame driver's face.</li> <li>Data for subjects with and without eyeglasses and sunglasses was used</li> </ul>	<ul style="list-style-type: none"> <li>Available online NTHU was used</li> <li>Adam method is used for CNN Optimization.</li> <li>Stander Haar cascade classifier implemented using OPEN CV with python.</li> </ul>	<ul style="list-style-type: none"> <li>96.42% accuracy was achieved</li> </ul>
Y. Ed-Doughmi <i>et al.</i> , 2020 [29]	<ul style="list-style-type: none"> <li>Drowsiness detection system based on CNN-based Machine Learning.</li> <li>Data for subjects with and without eyeglasses was used</li> </ul>	<ul style="list-style-type: none"> <li>3D Convolutional Networks was used</li> <li>Keras framework with the Python programming language was used.</li> <li>mobile application.</li> <li>Available online NTHU was used</li> </ul>	<ul style="list-style-type: none"> <li>97% accuracy was achieved</li> </ul>
R. Jabbar <i>et al.</i> , 2020 [17]	<ul style="list-style-type: none"> <li>Driver drowsiness and object distraction detection system using facial landmarks</li> </ul>	<ul style="list-style-type: none"> <li>Comparative analysis of four pertained models MLP-Model,</li> <li>CNN Model, VGG-16, AlexNet.</li> <li>Available online NTHU was used</li> <li>Dlib library is used to extract landmark coordination from images</li> <li>Android mobile application, Java Native Interface was used for implantation</li> </ul>	<ul style="list-style-type: none"> <li>Accuracy achieved by MLP, CNN, VGG-16, and AlexNet models were 80.9%, 83.3, 90.5, and 82. 8 respectively.</li> </ul>
Komal <i>et al.</i> , 2020 [11]	<ul style="list-style-type: none"> <li>Driver drowsiness detection based on eye state. CNN hyperparameters with Viola-Jones object detection.</li> <li>Subjects included with and without eyeglasses</li> </ul>	<ul style="list-style-type: none"> <li>Single shot multi-box detector Object detection technique was used.</li> <li>Tensor flow deep learning framework is used to implement the model.</li> </ul>	<ul style="list-style-type: none"> <li>90% accuracy was achieved</li> </ul>
Proposed system	<ul style="list-style-type: none"> <li>Driver sleepiness identification method using CNNs to extract information from images and LSTM.</li> <li>Subjects included without eyeglasses</li> </ul>	<ul style="list-style-type: none"> <li>Systematic Hyperparameters optimization were conducted for the CNN.</li> <li>OPEN CV with python is used for implementing Haar cascade classifier</li> <li>Empirically determination of Haar cascade algorithm parameters to identify open and close eyes</li> <li>Logitech webcam, NVidia Jatson Nano, LCD monitor were used</li> </ul>	<ul style="list-style-type: none"> <li>97.98% accuracy was achieved</li> </ul>

## 5. CONCLUSION




A significant percentage of car accidents are traced back to sleeping while driving. To address this issue, this study proposed an intelligent driver drowsiness detecting system based CNN model. The system was developed systematically by using; Haar cascade algorithm to preprocess the images, optimization of the CNN hyperparameters for precise classification, and defining the drowsiness criteria. This study showed that systematic optimization of the CNN hyperparameters could tremendously increase the system's accuracy. Overall, promising results were achieved from the proposed model for identifying the drowsiness of the drivers. The experimental results performed shown that the model achieved 99.87%, 99.63%, and 97.98% accuracy for training, validation, and testing, respectively. The high accuracy levels achieved by the system indicate the probability and need to deploy such systems for real-time application. The main challenge to developing a comprehensive real-time drowsiness detecting system seems unaccomplished yet. Even though our system demonstrated high accuracy compared with other developed systems, we could still add additional features to increase its reliability and practicality. Accordingly, the scope of our future work is divided into two areas; first, including an extra feature in drowsiness identification such as Yawning facial landmarks and abnormal head movement as well as night detection. Secondly, real-time implementation of the system.

## REFERENCES




- [1] "Global status report on road safety 2018," 2018. Accessed: Aug. 08, 2020. [Online]. Available: [https://www.who.int/violence\\_injury\\_prevention/road\\_safety\\_status/2018/en/](https://www.who.int/violence_injury_prevention/road_safety_status/2018/en/).
- [2] "Drowsy driving 2015." Accessed: Aug. 08, 2020. [Online]. Available: <https://crashstats.nhtsa.dot.gov/Api/Public/ViewPublication/812446>.
- [3] K. Al Hammadi, M. Ismaeel, and T. Faisal, "Intelligent car safety system," in *2016 IEEE Industrial Electronics and Applications Conference (IEACon)*, Nov. 2016, pp. 319–322, doi: 10.1109/IEACon.2016.8067398.
- [4] A. Picot, S. Charbonnier, and A. Caplier, "On-line detection of drowsiness using brain and visual information," *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, vol. 42, no. 3, pp. 764–775, May 2012, doi: 10.1109/TSMCA.2011.2164242.
- [5] J. I. Pilataxi, W. M. Vinan, and D. C. Garcia, "Design and implementation of a driving assistance system in a car like robot when fatigue in the user is detected," in *2014 IEEE ANDESCON*, Oct. 2014, pp. 1–1, doi: 10.1109/ANDESCON.2014.7098558.
- [6] C. Couprie, C. Farabet, L. Najman, and Y. LeCun, "Indoor semantic segmentation using depth information," in *1st International Conference on Learning Representations, ICLR 2013 - Conference Track Proceedings*, Jan. 2013, pp. 1–8, [Online]. Available: <http://arxiv.org/abs/1301.3572>.
- [7] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 8691, no. 3, pp. 346–361, Jun. 2014, doi: 10.1007/978-3-319-10578-9\_23.
- [8] Q. Wang, C. Rasmussen, and C. Song, "Fast, deep detection and tracking of birds and nests," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 10072 LNCS, Springer, Cham, 2016, pp. 146–155.
- [9] D. H. Hubel and T. N. Wiesel, "Receptive fields and functional architecture of monkey striate cortex," *The Journal of Physiology*, vol. 195, no. 1, pp. 215–243, Mar. 1968, doi: 10.1113/jphysiol.1968.sp008455.
- [10] K. Fukushima and S. Miyake, "Neocognitron: a new algorithm for pattern recognition tolerant of deformations and shifts in position," *Pattern Recognition*, vol. 15, no. 6, pp. 455–469, Jan. 1982, doi: 10.1016/0031-3203(82)90024-3.
- [11] Komal, P. Sharma, A. Lamba, B. Nagpal, and S. Chauhan, "Driver drowsiness detection system using convolutional neural network," *International Journal on Emerging Technologies*, vol. 11, no. 3, pp. 502–506, 2020.
- [12] H. B. Vanjani and U. Varyani, "Identify dozyneess of person using deep learning," *International Journal of Applied Engineering Research*, vol. 14, no. 4, pp. 845–848, 2019, [Online]. Available: <http://www.ripublication.com>.
- [13] T. H. Vu, A. Dang, and J.-C. Wang, "A deep neural network for real-time driver drowsiness detection," *IEICE Transactions on Information and Systems*, no. 12, pp. 2637–2641, Dec. 2019, doi: 10.1587/transinf.2019EDL8079.
- [14] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2001, vol. 1, pp. 1-511-I–518, doi: 10.1109/CVPR.2001.990517.
- [15] R. Jabbar, K. Al-Khalifa, M. Kharbeche, W. Alhajyaseen, M. Jafari, and S. Jiang, "Real-time driver drowsiness detection for android application using deep neural networks techniques," *Procedia Computer Science*, vol. 130, pp. 400–407, Jan. 2018, doi: 10.1016/j.procs.2018.04.060.
- [16] M. Omidyeganeh *et al.*, "Yawning detection using embedded smart cameras," *IEEE Transactions on Instrumentation and Measurement*, vol. 65, no. 3, pp. 570–582, Mar. 2016, doi: 10.1109/TIM.2015.2507378.
- [17] R. Jabbar, M. Shinoy, M. Kharbeche, K. Al-Khalifa, M. Krichen, and K. Barkaoui, "Driver drowsiness detection model using convolutional neural networks techniques for android application," *2020 IEEE International Conference on Informatics, IoT, and Enabling Technologies, ICIoT 2020*, pp. 237–242, 2020, doi: 10.1109/ICIoT48696.2020.9089484.
- [18] V. Chirra, S. ReddyUyyala, and V. KishoreKolli, "Deep CNN: a machine learning approach for driver drowsiness detection based on eye state," *Revue d'Intelligence Artificielle*, vol. 33, no. 6, pp. 461–466, Dec. 2019, doi: 10.18280/ria.330609.
- [19] R. Ghoddoosian, M. Galib, and V. Athitsos, "A realistic dataset and baseline temporal model for early drowsiness detection," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, Apr. 2019, pp. 178–187, doi: 10.1109/CVPRW.2019.00027.
- [20] W. Guo, L. Huang, and L. Liang, "A weld seam dataset and automatic detection of welding defects using convolutional neural network," in *Advances in Intelligent Systems and Computing*, vol. 905, Springer, Cham, 2020, pp. 434–443.
- [21] D. Sirohi, N. Kumar, and P. S. Rana, "Convolutional neural networks for 5G-enabled intelligent transportation aystem: a systematic review," *Computer Communications*, vol. 153, pp. 459–498, Mar. 2020, doi: 10.1016/j.comcom.2020.01.058.
- [22] R. Yamashita, M. Nishio, R. K. G. Do, and K. Togashi, "Convolutional neural networks: an overview and application in radiology," *Insights into Imaging*, vol. 9, no. 4, pp. 611–629, Aug. 2018, doi: 10.1007/s13244-018-0639-9.
- [23] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998, doi: 10.1109/5.726791.
- [24] T. Faisal, M. N. Taib, and F. Ibrahim, "Neural network diagnostic system for dengue patients risk classification," *Journal of Medical Systems*, vol. 36, no. 2, pp. 661–676, Apr. 2012, doi: 10.1007/s10916-010-9532-x.
- [25] N. N. A. Mangshor, I. A. A. Majid, S. Ibrahim, and N. Sabri, "A real-time drowsiness and fatigue recognition using support vector machine," *IAES International Journal of Artificial Intelligence (IJ-AI)*, vol. 9, no. 4, pp. 584–590, Dec. 2020, doi: 10.11591/ijai.v9.i4.pp584-590.
- [26] Q. A. Abed, O. Mohammed Fadhil, and W. L. Al-Yaseen, "Data mining in web personalization using the blended deep learning model," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 20, no. 3, pp. 1507–1512, Dec. 2020, doi: 10.11591/ijeecs.v20.i3.pp1507-1512.
- [27] R. I. Farhan, A. T. Maolood, and N. F. Hassan, "Performance analysis of flow-based attacks detection on CSE-CIC-IDS2018 dataset using deep learning," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 20, no. 3, pp. 1413–1418, Dec. 2020, doi: 10.11591/ijeecs.v20.i3.pp1413-1418.
- [28] K.-A. Kwon *et al.*, "High-speed camera characterization of voluntary eye blinking kinematics," *Journal of The Royal Society Interface*, vol. 10, no. 85, Aug. 2013, doi: 10.1098/rsif.2013.0227.
- [29] Y. Ed-Doughmi, N. Idrissi, and Y. Hbali, "Real-time system for driver fatigue detection based on a recurrent neuronal network," *Journal of Imaging*, vol. 6, no. 3, Mar. 2020, doi: 10.3390/jimaging6030008.

## BIOGRAPHIES OF AUTHORS






**Dr. Tarig Faisal**    received his master degree of Mechatronics Engineering from IIUM university in 2006 and his Ph.D. degree in Signal Processing from University of Malaya, Malaysia, in 2011. He has been the Dean of Academic Operations at Higher Colleges of Technology since 2018. He has published more than 35 research articles with more than 400 citations and H-index of 11. His research focuses on biomedical signal processing, intelligent systems, robotics, control, embedded system design, IoTs, machine learning, and outcome based education. He has been a reviewer for multiple journals including IEEE, Elsevier, Taylor & Francis, and Springer Nature. He has more than 20 years of academic and industry experience of which he worked as an engineering, assistant professor, Programs Chair, Head of department, and Division Chair. He is also a chartered engineering as well as Senior Fellow of Higher Education Academy. Dr. Tarig can be contacted at email: tfaisal@hct.ac.ae.






**Isaias Negassi**    is alumni from the Electrical Engineering Department of Higher Colleges of Technology, UAE. Isaias has been working as research assistant in the area of embedded system design and Artificial intelligence. He has won many awards including 1<sup>st</sup> prize on Engineering Project Competition in HCT AL Dhafrah colleges in 2017, 2<sup>nd</sup> prize on Projects and innovative idea competition 2018, 1<sup>st</sup> prize on AL Dhafrah innovation Carnival competition 2019, and Bronze Award in HCT Ibtikar Innovation and Entrepreneur Competition 2020. He can be contacted at email: Isaiastesfu27@gmail.com.






**Ghebrehiwet Goitom**    is alumni from the Electrical Engineering Department of Higher Colleges of Technology, UAE. Ghebrehiwet has been working as research assistant in the area of embedded system design and intelligent system. He has won many awards including 1<sup>st</sup> prize on Engineering Project Competition in HCT AL Dhafrah colleges in 2017, 2<sup>nd</sup> prize on Projects and innovative idea competition 2018, 1<sup>st</sup> prize on AL Dhafrah innovation Carnival competition 2019, and Bronze Award in HCT Ibtikar Innovation and Entrepreneur Competition 2020. He can be contacted at email: gebrishfet@gmail.com.






**Mohammed Yassin**    is alumni from the Electrical Engineering Department of Higher Colleges of Technology, UAE. Mohammed has been working as research assistant in the area of embedded system design and intelligent system. He has won many awards including 1<sup>st</sup> prize on Engineering Project Competition in HCT AL Dhafrah colleges in 2017, 2<sup>nd</sup> prize on Projects and innovative idea competition 2018, 1<sup>st</sup> prize on AL Dhafrah innovation Carnival competition 2019, and Bronze Award in HCT Ibtikar Innovation and Entrepreneur Competition 2020. He can be contacted at email: mohammedyssn2019@gmail.com.



**Anees Bashir**    received Bachelor in electronics and communication engineering from Anna University, India in 2012, and master degree in applied electronics engineering from M.S. University India, in 2014. Her main research interests include embedded system design, signal processing, outlier detection, data mining. She is currently working on her research based on Moving-horizon strategy for anomaly detection in discrete LTI systems. She can be contacted at email: abashir1@hct.ac.ae.



**Dr. Moath Awawdeh**    (Member, IEEE) received the B.S. in communication and software engineering from Al Balqa Applied University for Engineering Technology, Jordan, in 2009, and Ph.D. in mathematical engineering and simulation with specialization in control engineering from the University of Genoa, Italy, in 2014. He is currently an applied research coordinator at Higher Colleges of Technology. Beside his active work as a reviewer for different international journals, He is on the Editorial Board of ASET conference and JAETS journal. His main research interests include outlier detection, data mining, machine learning, signal processing, and control engineering applications. He is currently working on the problem of outlier detection and correction in LTI systems with a specific approach of state estimation and measurements validation. He can be contacted at email: mawawdeh@hct.ac.ae.