ECG-Based Driver’s Stress Detection Using Deep Transfer Learning and Fuzzy Logic Approaches

MUHAMMAD AMIN1,2, KHALIL ULLAH3, MUHAMMAD ASIF1, ABDUL WAHEED4,5, SANA UL HAQ1, MAHDI ZARREEI6, (Senior Member, IEEE), AND R. R. BISWAL6

1Department of Electronics, University of Peshawar, Peshawar 25120, Pakistan
2Department of Computer Science, Iqra National University, Peshawar 25000, Pakistan
3Department of Software Engineering, University of Malakand, Dir Lower, Chakdara 23050, Pakistan
4Department of Computer Science, Northern University, Nowshera 24100, Pakistan
5School of Electrical and Computer Engineering, Seoul National University, Seoul 08826, South Korea
6School of Engineering and Sciences, Tecnológico de Monterrey, Zapopan 45201, Mexico

Corresponding authors: Abdul Waheed (abdul@netlab.snu.ac.kr) and Mahdi Zareei (m.zareei@tec.mx)

ABSTRACT Driver’s stress detection is a critical research area that helps reduce the likelihood of traffic accidents and driver’s health complexities due to prolonged stress. Previous work in this area is heavily based on traditional machine learning models that classify the driver’s stress levels using handcrafted features extraction techniques. Extracting the best features using these approaches is always a challenging task. Recently, deep learning techniques have emerged for constructing reliable features automatically and classifying the classes with high accuracy. However, large deep learning models face gradient exploding or vanishing problems. Moreover, acquiring a large dataset for training an entire network from scratch is also a challenging task. This paper is based on the deep transfer learning technique to avoid these problems and to reduce computational cost and time. Seven models are proposed for real-world driver’s stress levels detection using Electrocardiogram (ECG) signals. Different Convolutional Neural Network (CNN)-based pre-trained networks are used to classify the driver’s three stress levels. The time-frequency ECG components for the three stress levels are obtained as scalogram images using a normalized Continuous Wavelet Transform (CWT) filter bank and Morse wavelet. Results show that Model 5 based on Xception outperforms the GoogLeNet, DarkNet-53, ResNet-101, InceptionResNetV2, DenseNet-201, and InceptionV3 based models by 11.32%, 11.32%, 9.45%, 7.54%, 5.66%, and 1.88% respectively and achieves 98.11% overall validation accuracy. Ranking estimation using fuzzy logic approach shows that Xception based Model 5 achieves the highest rank for driver’s high and medium stress levels, while DenseNet-201 based Model 4 achieves the highest rank for low-stress level detection among the other models.

INDEX TERMS Driver’s stress detection, ECG signals, signal processing, continuous wavelet transform, deep transfer learning, pre-trained networks, fuzzy logic.

I. INTRODUCTION
Driving is among the most unhealthy professions in the world [1], [2]. Due to the complex and hazardous nature, effective driving activities always need full utilization of both physical and mental skills [3]. In case of any hazardous driving situation, the driver’s mind triggers an acute stress response due to increased activity in the sympathetic nervous system, which may stop the body from taking appropriate action timely and may lead to severe losses [4]–[6]. Professional drivers are more prone to stress than workers in other sectors [7], [8]. Stress negatively impacts driving behavior, which may often lead to traffic accidents, thus causing significant damage to humans and vehicles every year [9], [10]. In addition, prolonged stress may also raise the risk of cardiac, gastrointestinal, and mental diseases [11], [12]. Hazardous situations may occur due to several factors, including human errors, individual conditions, and environmental conditions [13]. About 90% of traffic accidents are caused due to driving errors. Human errors due to inattention, distraction, mental workload, and poor observation are most frequent (41%) [14]. Internal or individual conditions include the driver’s mood, personality, age, gender, and crash history. Environmental conditions are based on weather, visibility,
driver-environment interactions, and driving routes. Human errors are primarily associated with drivers’ affective states; thus, a thorough understanding of these states is necessary to avoid any disaster.

Driver’s stress levels can be detected using physiological, physical, and contextual data [15]. The autonomic nervous system controls physiological responses through the sympathetic and parasympathetic nervous systems. Among several other physiological signals, Electrocardiogram (ECG), galvanic skin response (GSR), and respiration (RESP) signals are most frequently used in the study of driver’s stress levels detection. Different physiological signals can be fused to detect the driver’s stress levels more effectively [16]. The physical responses are controlled by the somatic motor system in the form of skeleton, muscles, and tissues movements. Physical data include vehicle dynamics data, facial expressions, and speech. The dynamic vehicle data has shown a strong correlation with drivers’ stress levels [17]. Contextual data such as the driver’s individual and ambient parameters correlate to the driver’s stress levels [18]. Multimodal data, including physiological signals and other information regarding the driver, vehicle, and environment, can enhance the driver’s stress levels classification [15].

Previous work in this area is greatly based on traditional machine learning algorithms to classify driver’s stress levels. Healey and Picard [16], Zhang et al. [20], Chen et al. [4], Haouij et al. [21], Lopez-Martinez et al. [22], Vargas-Lopez et al. [23], Dalmeida and Masala [24], Lee et al. [17], Bianco et al. [25], and Zontone et al. [26] have proposed driver’s stress detection models in real world and simulated driving scenarios. However, extracting the best handcrafted features from different physiological and physical data used these models is always a challenging task. Moreover, different linear and non-linear techniques used in these models had not been able to accomplish the robust analysis of such data [29].

To overcome these problems, deep learning models have been employed to automatically construct complex nonlinear features reliably [31], [33]. These models also provide noise robustness and improved classification accuracy [29]. Various researcher including Lee et al. [37], Hajinoroozi et al. [38], Yan et al. [39], and Rastgoo et al. [15] have proposed CNN-based models for driver’s state detection using different modalities. However, large deep learning models face gradient exploding or vanishing problems. Moreover, acquiring a large dataset for training an entire CNN form scratch is also a challenging task. Deep transfer learning is a recently emerging approach for developing accurate models very quickly [39]. In deep transfer learning, a pre-trained network already trained on a large benchmark dataset is used for training the target network using a relatively small target dataset with reduced computational cost and time. Xing et al. [40] used body images to monitor the driver’s seven common activities based on deep transfer learning.

In this research work, the real-world driver’s stress is classified into three (low, medium, high) stress levels using deep transfer learning. The seven proposed models are assessed on the ECG signals collected from the PhysioNet public database (Stress Recognition in Automobile Drivers) [19]. Signals are extracted from 9 drivers during 14 drives each comprising of the rest, highway, and city scenarios to cause various stress levels. The proposed models are based on the pre-trained networks including GoogLeNet, DarkNet-53, ResNet-101, InceptionResNetV2, Xception, DenseNet-201, and InceptionV3. The time-frequency ECG components for the three stress levels are obtained as scalogram images using a normalized Continuous Wavelet Transform (CWT) filter bank and Morse wavelet. A fuzzy logic-based ranking estimation approach is used for the performance evaluation of the proposed models. The performance of the proposed models is compared with existing traditional machine learning and deep learning models.

Contributions of the current work include:

1) To transform the 1D ECG signals obtained for driver’s low, medium, and high stress levels into 2D time-frequency images (scalograms) using CWT techniques,
2) To propose efficient and accurate models based on the latest available pretrained networks (GoogLeNet, DarkNet-53, ResNet-101, InceptionResNetV2, Xception, DenseNet-201, and InceptionV3) for classifying the driver’s three stress levels using the 2D scalogram images of the ECG signals obtained from the PhysioNet public database [19],
3) To estimate the ranking of the proposed models using the appraisal scores (λ) calculated for the driver’s low, medium, and high stress levels using fuzzy logic based EDAS approach for different performance metrics including accuracy, sensitivity, precision, F-Score, and specificity.

The paper is organized into six sections. Section II includes traditional machine learning approaches for stress level classification in different scenarios, including real-time and simulated driving conditions. This section also discusses the current deep learning classification based automatic features extraction and classification approaches for driver’s behavior and cognitive states. The proposed transfer learning-based approach is briefly highlighted in this section. Section III is about the proposed methodology, which elaborates the dataset used, preprocessing, and wavelet transform of the ECG signals. It also discusses different pre-trained networks and their fine-tuning to the target ECG dataset. Then, the networks’ training is explained in terms of image resizing, augmentation mechanism, and parameters setting. The classification metrics and fuzzy logic-based methods are also defined in this section. Section IV includes the overall and level-wise classification results. The performance evaluation using fuzzy logic-based ranking estimation is also used in this section. Discussions about the achieved results and comparison with other schemes are given in.
Section V. Finally, Section VI concludes the paper and provides the future perceptions to investigate this area further.

II. RELATED WORK

This section reviews the previous work on the driver’s stress analysis and highlights the current contribution to this field. Several approaches in driver’s stress detection for the real world and simulated driving conditions exist in the literature. Healey and Picard [16], Zhang et al. [20], Chen et al. [4], Haouij et al. [21], Lopez-Martinez et al. [22], Vargas-Lopez et al. [23], and Dalmeida and Masala [24] proposed real-world driver’s stress detection models using either single or fusion of different physiological signals including ECG, Heart Rate (HR), electromyogram (EMG), GSR, and RESP signals acquired from the PhysioNet real-world driving database [19]. Similarly, Lee et al. [17], Bianco et al. [25], and Zontone et al. [26] have proposed driver’s stress detection models for simulated driving conditions. The former study is based on the physiological signals and physical information acquired during the experimental work, while the latter two studies have used physiological signals acquired from the Scientific Data public database [27] and self-dataset not publicly available. All these studies are based on handcrafted feature extraction techniques, and different traditional machine learning algorithms were used to classify drivers’ stress levels. Although handcrafted features produce encouraging results, extracting the best features using these approaches is always challenging as the quality of extracted features significantly affects the classification performance [30]. Moreover, these approaches are inadequate for capturing the nonlinear correlation across different signals which appear simultaneously [32]. Different non-linear techniques used in these approaches had not been able to accomplish the robust analysis of such complex signals [29]. These approaches are time-consuming, ad-hoc, less robust to noise, and need expert knowledge [34].

To overcome these problems, deep learning techniques have been developed to construct reliable features automatically [31], [33]. These techniques learn and classify raw data using neural networks with multilayers [35]. Besides automatic feature learning from raw data, deep learning models provide noise robustness and improved classification accuracy [29]. Several authors including Lee et al. [37], Hajinooroozi et al. [38], Yan et al. [39], Kanjo et al. [36], and Rastgoo et al. [15] have proposed models for driver’s state detection based on deep learning approaches. Lee et al. [37] proposed a CNN-based driver’s stress detection approach using continuous recurrence plots of three physiological signals acquired from the PhysioNet database [19]. The proposed scheme has achieved 95.67% accuracy, which is relatively small for two stress classes. Similarly, Hajinooroozi et al. [38] performed the driver’s cognitive states estimation using Electroencephalogram (EEG) signals in simulated driving conditions. The proposed channel-wise CNN outperformed other algorithms by achieving 86.08% accuracy. In another study, Yan et al. [39] presented an adapted region-based CNN (R*CNN) model for driver’s behavior recognition using the image’s pose information and contextual cues. It outperformed the traditional machine learning algorithms by achieving a mean average precision of 97.76% for six classes. In a hybrid approach, Kanjo et al. [36] presented a CNN and Long Short-term Memory (LSTM) based model for emotion classification using multimodal data. The proposed model has outperformed Multilayer Perceptron (MLP) by achieving a maximum accuracy of 94.7% for four classes. Yet another hybrid multimodal fusion scheme model was presented by Rastgoo et al. [15] using CNN and LSTM for stress levels detection during simulated driving. The proposed model outperformed the traditional machine learning-based models by achieving an average accuracy of 92.8%. Despite the significance of traditional deep learning approaches, large deep learning models face gradient exploding or vanishing problems. Moreover, acquiring a large dataset for training an entire CNN from scratch is also a challenging task.

Deep transfer learning is a recently emerging approach for developing accurate models in a fast way [39]. In deep transfer learning, a pre-trained network on a large benchmark dataset is used to train the target network using a relatively small target dataset with reduced computational cost and time. Xing et al. [40] used body images to monitor the driver’s seven activities based on deep CNN and transfer learning. Among the three different pre-trained CNN models, AlexNet achieved the maximum classification accuracy of 81.6% for seven classes. This paper uses different pre-trained networks to classify the driver’s three stress levels using ECG signals. Moreover, a fuzzy logic-based ranking estimation approach is used for the performance evaluation of the proposed models. Basar et al. [42], [44], and Mehmood et al. [43], used the EDAS approach for performance comparison of different schemes. The EDAS approach is used in the current work to estimate the proposed models based on the appraisal scores (λ) calculated for the driver’s low, medium, and high-stress levels using different performance metrics.

III. METHODOLOGY

This section gives a brief overview of the proposed deep transfer learning architecture and the public real-time driving database for detecting drivers’ stress levels. The proposed deep transfer learning architecture is implemented in MATLAB. Several CNN-based pre-trained networks are used to train the models, such as GoogLeNet, InceptionV3, ResNet-101, DenseNet-201, Xception, InceptionResNetV2, and DarkNet-53. PhysioNet [19] public real-time driving database is used for analyzing the proposed model. The System architecture is given in Figure 1. Five widely used classification metrics are used to assess the overall performance of the proposed models. These performance metrics are defined in section III-E. Moreover, the EDAS method is used to assess...
the performance of the proposed models for the three stress levels. The EDAS method is discussed in subsection III-F.

A. DATASET DESCRIPTION

ECG signals are acquired from the PhysioNet public database (Stress Recognition in Automobile Drivers) [19]. A modified Volvo S70 series station wagon was used to collect the data using four physiological sensors, i.e., ECG, GSR, EMG, and RESP connected to an embedded computer. Experiments were conducted using 9 drivers. The database consists of 14 drives with complete data and clear markers, which are used in this study.

All drives consisted of a 31 km fixed route in the Boston area. Each drive consisted of rest, highway, and city driving periods which were presumed to produce low, medium, and high-stress levels, respectively. At the start, drivers were briefed about the route map and guided for consistent driving, e.g., observing the speed limits and avoiding listening to the radio. To avoid heavy highway traffic, drives were conducted in the middle of the morning or afternoon. Each drive included two 15-minutes rest periods at the start and end to gather baseline quantities for low-stress conditions. The main street city driving started after leaving the garage through side streets, indicating high-stress conditions due to stop-and-go traffic. The route then led to continuous highway driving, after passing the toll, indicating a medium-stress condition. To reach back to the starting point, all the above periods were traveled in the reverse direction. The total length of the drive with two 15-minutes rest periods was 50 to 90 minutes.

This study considered only the ECG signal due to its close correlation with the driver stress level [16]. Modified lead II configuration was used for electrodes placement to minimize the motion artifacts. Although 17 drives are available in the database, only 14 drives (drive02, drive04, drive05, drive06, drive07, drive08, drive09, drive10, drive11, drive12, drive13, drive14, drive15, and drive16) were used because of the missing labels in the remaining drives. Figure 2 shows the variations in QRS waves peaks for randomly selected ECG signals from high, low, and medium stress segments.

B. PRE-PROCESSING

ECG signals acquired from the human body are generally low amplitude and mostly contaminated by base-line wander, power-line interference (PLI), electromyographic interference, and some high frequency noises [45]. Baseline wander is a varying low frequency (0.15 − 0.5 Hz) signal induced by breathing, muscle movement, electrode misplacement, and impedance of skin and electrode. PLI of 50 Hz and its harmonics are also added into ECG signals due to the AC-coupling of electrodes with power
C. TIME-FREQUENCY REPRESENTATION OF ECG SIGNAL

ECG signal comprises several time-frequency components which are helpful in driver’s stress level classification. The ECG signal is first transformed into a time-frequency domain using CWT to extract these components as a scalogram. The CWT for a signal \( x(t) \) can be defined as:

\[
C_x(a, b) = \frac{1}{\sqrt{a}} \int_{-\infty}^{+\infty} x(t) \psi \left( \frac{t-b}{a} \right) dt
\]  

where \( \psi(t) \) is the wavelet function, \( a \) and \( b \) are the scale and translation parameters, respectively.

The wavelet coefficients (or scalograms) of the signal are obtained by applying a wavelet function “\( \psi(t) \)” scaled and shifted in time using the scale “\( a \)” and translation “\( b \)” parameters of CWT. The scale parameters provide both low-frequency high-time and high-frequency low-time resolution to detect high and low-frequency events, respectively [50].

The frequency-domain representation of the scale parameter is given below:

\[
F = \frac{f_C f_S}{a}
\]  

Here, \( f_C \) is the mother wavelet’s center frequency, and \( f_S \) is the signal \( x(t) \) sampling frequency. A CWT filter bank is created to get the scalogram images for the low, medium, and high-stress classes. The peak magnitude of all the passbands filters is approximately set to a value of 2 after the normalization. The ECG signals were segmented into 6 seconds with a total of 2976 segments. Various wavelet functions (or mother wavelets) are available in CWT, which are carefully selected better to decompose a particular signal into the time-frequency domain. This paper uses the Morse wavelet as the mother wavelet because of its similarity with the ECG signal. The time-domain representation of the Morse wavelet is given below:

\[
\psi_{\beta, \gamma} (t) = \frac{1}{2\pi} \int_{0}^{\infty} a_{\beta, \gamma} e^{i \omega t} e^{-\alpha \omega^2} \phi_{\beta, \gamma} d\omega
\]

Here, \( a_{\beta, \gamma} \) is the normalizing constant, \( \gamma \) characterizes the Morse wavelet’s symmetry, and \( \beta \) is the compactness parameter. The scales are approximately set to 12 wavelet band-pass filters per octave (12 voices per octave). Due to the highest-frequency passband design, the magnitude falls to half of the peak value at the Nyquist frequency. To present a more accurate interpretation of the ECG signal, L1 normalization is used in CWT to achieve equal magnitude for equal amplitude oscillatory components of the ECG signal at different scales. The amplitudes of the oscillatory components of the ECG signal must agree with the amplitudes of the subsequent scalograms. Figure 3 shows three scalogram images created to get the scalogram images for the low, medium, and high-stress classes. The peak magnitude of all the passbands filters is approximately set to a value of 2 after the normalization. The ECG signals were segmented into 6 seconds with a total of 2976 segments. Various wavelet functions (or mother wavelets) are available in CWT, which are carefully selected better to decompose a particular signal into the time-frequency domain.

D. TRANSFER LEARNING

Transfer learning is a fast way of developing accurate models [51]. Acquiring a large dataset for training an entire CNN from scratch is generally a challenging task. Thus, transfer learning is the best choice, where a pre-trained network already trained on a large benchmark dataset (such as ImageNet) is used for solving the different problems with reduced computational cost. Features extracted by a primary (pre-trained) network are then transferred to a target network for training using a target dataset [52]. Fine-tuning of the primary network depends upon the size and similarity of the datasets. If the target network’s dataset is small and different from the primary network’s dataset, training some layers and leaving others frozen would be a good strategy. For a small dataset with a large number of parameters, more layers may be kept frozen.
frozen to avoid overfitting. Moreover, it may also be helpful to use a data augmentation technique.

1) PRE-TRAINED NETWORKS

The work in this paper is based on scalogram images acquired from ECG signals. Although such images are relatively different from ordinary images, the primary network trained on a vast dataset such as ImageNet can effectively recognize them [53].

Traditional CNN provides gradual connections among different layers and may not expend much due to gradient exploding or vanishing problems. Several CNN-based pre-trained networks are used to train the models, such as GoogLeNet, InceptionV3, ResNet-101, DenseNet-201, Xception, InceptionResNetV2, and DarkNet-53. These networks are trained on ImageNet to classify images into 1000 object categories. GoogLeNet, ResNet-101, and DenseNet-201 have an image input size of 224 × 224 × 3, InceptionV3, Xception, and InceptionResNetV2 have an image input size of 299 × 299 × 3, while DarkNet-53 has an image input size of 256 × 256 × 3. The pre-trained networks are fine-tuned to the driver’s stress level detection dataset.

GoogLeNet is based on a relatively shallower design having only 22 deep layers [54]. It uses inception modules to achieve computational efficiency, reduce dimensionality in deeper networks, and deal with overfitting problems. After convolution and max-pooling of various varying size filters, the concatenated result is sent to the next inception layer. Auxiliary classifiers are added to the network to achieve discrimination and additional regularization.

InceptionV3 is an extended version of GoogLeNet containing 48 deep layers [55]. Unlike GoogLeNet, InceptionV3 is augmented with factorized convolutions, regularization, dimension reduction, and parallelized computations techniques. Several distinct convolutional filters are concatenated into a single filter to reduce the number of training parameters and computational complexity.

ResNet-101 with 101 layers was introduced to solve the gradient backpropagation issues in denser networks [56]. Deeper networks may suffer from degradation problems during the convergence process, which triggers the accuracy to reduce quickly. ResNet-101 is based on the residual connection, which skips some layers to reduce degradation [57].

DenseNet-201 is a variant of ResNet having 201 deep layers [58]. It provides information and gradients flow mechanisms throughout the network to counter the gradient vanishing or exploding problems. The regularizing effect of the dense connections reduces the overfitting problem for small size training sets. DenseNet-201 uses a feature reuse mechanism to produce condensed models which are easy to train and highly parameter-efficient. Feature maps learned by different layers are concatenated to produce distinct input for coming layers to improve efficiency.

ResNet-101 is a residual network having 101 deep layers [56]. Deeper networks may suffer from degradation problems during the convergence process, which triggers the accuracy to reduce quickly. ResNet uses a deep residual learning framework that addresses the degradation problem [57].

Xception is an extreme version of Inception, having 71 deep layers [59]. Unlike Inception, the Xception architecture replaces the Inception modules with a linear stack of depth-wise separable convolution layers. Linear residual connections can be used to improve convergence speed and classification performance.
Inception-ResNetV2 is a substantially deeper variation of InceptionV3, having 164 deep layers [60]. It combines Residual connections and simplified Inception blocks to avoid the degradation problem and achieve better training performance in reduced time.

DarkNet-53, which is the backbone of the You Only Look Once (YOLOv3) network, consists of 53 deep layers [61]. It has different size convolutional filters with residual connections like the ResNet, which easily control the gradient propagate through deeper layers and perform network training.

2) NETWORK’S FINE-TUNING

The target dataset for driver’s stress level detection using ECG signal is small, so the primary networks are fine-tuned accordingly. The primary networks are already trained on a vast dataset such as ImageNet. A target network is designed that copy all the primary network’s layers except the output layer. This includes the source network’s parameters containing knowledge learned from the primary dataset and relevant to the target dataset. The target network’s output layer changes according to the target dataset’s classes, and the network parameters are randomly initialized. The target network is then trained on the target dataset for the driver’s stress level detection. Figure 5 shows the concept of fine-tuning a pre-trained network to a small dataset.

Features extracted by the convolutional layers of the pre-trained network from the scalogram images are used by the last two layers (learnable and classification layers) to classify the driver’s stress level. The information held by these two layers is used to find the class probabilities, lose value, and predicted labels from the features extracted by the convolutional layers. These two layers are changed according to the driver’s stress level detection dataset. The last fully-connected layer holding the learnable weights is replaced by new output layers having three classes instead of 1000 classes. The classification layer’s name is changed to avoid conflicts with the original weights. Due to the small dataset, the whole convolutional base of the pre-trained networks is kept frozen, and the new output layers are trained from scratch. The weights of each layer are frozen by setting the learning rates of the respective layer to zero; thus, the parameters of these layers remain unchanged during the training process. Freezing the convolutional base can increase the training speed significantly and reduce the overfitting problem for small target datasets.

3) NETWORK’S TRAINING

The original scalogram image size is 100 × 2048 × 3, different from the image input size of different pre-trained networks. For example, GoogLeNet, ResNet-101, and DenseNet-201 have an image size of 224 × 224 × 3, InceptionV3, Xception, and InceptionResNetV2 have an image size of 299 × 299 × 3, while DarkNet-53 has an image size of 256 × 256 × 3. To automatically resize the images for pre-trained networks, an augmented image datastore is utilized. Moreover, some augmentation operations are randomly performed on the images, such as flipping, translation, and scaling. The training images are flipped vertically, translated up to 30 pixels, and scaled both in horizontal and vertical axis up to 10%.

Several parameters need to be set for the network’s training, such as training algorithm, mini-batch size, validation
frequency, initial learn rate, and several epochs. The training algorithm is set to stochastic gradient descent with momentum (SGDM). Mini-batch size is the number of images used per iteration. Due to the small dataset, it is set to a small value (20) to divide the target dataset equally and to ensure the usage of the entire dataset for each epoch. The validation frequency of the network is the number of iterations between evaluations of validation metrics. It is also set to a low value (10), due to the small target dataset. The initial learning rate for training is set to a moderate value of $3.5 \times 10^{-3}$ s to minimize learning in the target network’s layers. The maximum number of epochs is set to 25 with 24 training iterations per epoch. It is the complete training cycle on the whole target dataset.

### E. CLASSIFICATION METRICS

Classification metric is used to evaluate the discrete classes to find the best-performing model. The proposed models are evaluated based on five performance metrics, including accuracy (ACCY), sensitivity (SSY), precision (PRC), F-Score (FS), and specificity (SPCY). The range for all the five classification metrics is $[0, 1]$. Accuracy is the ratio of correctly predicted to all predicted instances as defined in the following line:

$$\text{ACCY}_k = \frac{TP_k + TN_k}{TP_k + TN_k + FP_k + FN_k}$$  \hspace{1cm} (4)

Here $TP_k$, $FN_k$, $TN_k$, and $FP_k$ represent true positive, false negative, true negative, and false positive for the $k_{th}$ class. $TP_k$ and $FP_k$ indicate the number of $k_{th}$ classes correctly predicted and the number of two other classes incorrectly predicted as the $k_{th}$ class respectively. $TN_k$ and $FN_k$ indicate that the number of two other classes not classified as the $k_{th}$ class and the number of $k_{th}$ classes classified as the two other classes respectively.

Sensitivity is the ratio of true positive to all positive instances in ground truth as defined in following equation:

$$\text{SSY}_k = \frac{TP_k}{TP_k + FN_k}$$  \hspace{1cm} (5)

Precision is the ratio true positive to all positive instances predicted by the classifier as given below mathematically:

$$\text{PRC}_k = \frac{TP_k}{TP_k + FP_k}$$  \hspace{1cm} (6)

F-Score is the harmonic mean between precision and sensitivity. It is used to measure the test’s accuracy. Mathematical definition of F-Score is given below:

$$\text{FS}_k = \frac{2TP_k}{2TP_k + FP_k + FN_k}$$  \hspace{1cm} (7)

Specificity is the correctly identified actual negative instances as defined in the following equation:

$$\text{SPCY} = \frac{TN_k}{TN_k + FP}$$  \hspace{1cm} (8)

### F. FUZZY LOGIC BASED APPROACH

EDAS is a fuzzy logic approach used for performance comparison of the driver’s stress level detection models. Seven driver’s stress level detection models are proposed in this work which is based on different state-of-the-art transfer learning approaches. EDAS method is used for ranking estimation of these stress detection models in terms of accuracy, sensitivity, precision, F-score, and specificity defined in section III-E. These five performance metrics are considered as criteria for the pre-trained networks to classify the driver’s stress level. First, the cross-efficient values ($\psi_{\beta}$) of various parameters of each of the proposed models are calculated for a particular driver’s stress level. Then the average positive ($P_3$) and negative distances ($N_5$) are calculated next. The appraisal scores ($\lambda$) are measured using the aggregate values of ($SP_3$) and ($SN_5$) for ranking estimation of the proposed models.

To estimate the ranking of different pre-trained Networks, EDAS is following the eight steps given below:

**Step 1:** This step is used to calculate the “solution of the average value ($\psi$)” of all matrices as given in the following equations:

$$\psi = [\psi_{\beta}]_{1 \times \delta}$$  \hspace{1cm} (9)

where

$$\psi_{\beta} = \frac{\sum_{i=1}^{\alpha} X_{a\beta}}{X}$$  \hspace{1cm} (10)

The aggregate calculation of Eqs. (9) and (10) can be obtained as the average value ($\psi$) for every criterion calculated values against each performance metric.

**Step 2:** In this step the positive distances from average ($P_3$) is determined using the following equation:

$$P_3 = [(P_3)_{a\beta}]_{\delta \times \delta}$$  \hspace{1cm} (11)
If the $\beta$th criterion is more helpful then $(P_{\beta})_{\alpha\beta}$ is computed using equation given below:

$$ (P_{\beta})_{\alpha\beta} = \frac{\text{Maximum}(0, (AV_{\beta} - X_{\alpha\beta}))}{AV_{\beta}} \quad (12) $$

If the criterion is not helpful, then $(P_{\beta})_{\alpha\beta}$ is determined mathematically in the following line:

$$ (P_{\beta})_{\alpha\beta} = \frac{\text{Maximum}(0, (X_{\alpha\beta} - AV_{\beta}))}{AV_{\beta}} \quad (13) $$

Here, $(P_{\beta})_{\alpha\beta}$ is the positive distance of $\beta$th model from the average value for $\alpha$th performance parameter. The average $(P_{\beta})$ is calculated for each proposed model on the basis of driver’s each stress level.

**Step 3:** In this step the negative distances from average $(N_{\beta})$ is computed using the following equation:

$$ (N_{\beta})_{\alpha\beta} = [(N_{\beta})_{\alpha\beta}]_{5 \times 5} \quad (14) $$

If the $\beta$th criterion is more helpful then negative distances from average $(N_{\beta})_{\alpha\beta}$ is calculated using equation given below:

$$ (N_{\beta})_{\alpha\beta} = \frac{\text{Maximum}(0, (AV_{\beta} - X_{\alpha\beta}))}{AV_{\beta}} \quad (15) $$

If the $\beta$th criterion is not helpful, then negative distances from average $(N_{\beta})_{\alpha\beta}$ is determined mathematically using the following equation:

$$ (N_{\beta})_{\alpha\beta} = \frac{\text{Maximum}(0, (X_{\alpha\beta} - AV_{\beta}))}{AV_{\beta}} \quad (16) $$

Here, $(N_{\beta})_{\alpha\beta}$ is the negative distance of $\beta$th model from the average value for $\alpha$th performance parameter. The average $(N_{\beta})$ is calculated for each proposed model based on the driver’s stress level.

**Step 4:** In this step, the weighted sum of $(P_{\beta})_{\alpha\beta}$ for driver’s stress detection model is determined mathematically using the following equation:

$$ (SP_{\beta})_{\alpha} = \sum_{\beta=1}^{x} y_{\beta}(P_{\beta})_{\alpha\beta} \quad (17) $$

The aggregate $(SP_{\beta})$ is calculated for each proposed model based on the driver’s each stress level.

**Step 5:** In this step, the weighted sum of $(N_{\beta})_{\alpha\beta}$ for the driver’s stress detection model is calculated using equation given below:

$$ (SN_{\beta})_{\alpha} = \sum_{\beta=1}^{x} y_{\beta}(N_{\beta})_{\alpha\beta} \quad (18) $$

The aggregate $(SN_{\beta})$ is calculated for each proposed model based on the driver’s each stress level.

**Step 6:** This step is used to normalize and calculate the scores of $(SP_{\beta})_{\alpha}$ and $(SN_{\beta})_{\alpha}$ for driver’s stress detection model using the following two equations:

$$ N(SP_{\beta})_{\alpha} = \frac{(SP_{\beta})_{\alpha}}{\text{maximum}_{\alpha}(SP_{\beta})_{\alpha}} \quad (19) $$

$$ N(SN_{\beta})_{\alpha} = 1 - \frac{(SN_{\beta})_{\alpha}}{\text{maximum}_{\alpha}(SN_{\beta})_{\alpha}} \quad (20) $$

**Step 7:** This step is used to calculate the appraisal score ($\lambda$) for the driver’s stress detection model on the basis of the aggregate scores of $N(SP_{\beta})_{\alpha}$ and $N(SN_{\beta})_{\alpha}$ using the following equation:

$$ \lambda_{\alpha} = \frac{1}{2} \left( N(SP_{\beta})_{\alpha} - N(SP_{\beta})_{\alpha} \right) \quad (21) $$

The range of appraisal score ($\lambda$) is given as $0 \leq \lambda_{\alpha} \leq 1$.

**Step 8:** This step is used to determine the appraisal scores ($\lambda$) in decreasing order and then perform the ranking estimation of the proposed models for the driver’s stress level. The best ranking model is the one having the lowest appraisal scores ($\lambda$).

**IV. RESULTS**

The ECG signals for different stress segments of 14 individual drives are converted to scalogram images which are then stored in a single directory for classification. The images are randomly divided into two groups with 90% and 10% split for training and validation respectively using the random seeds. To assess the performance of the seven proposed driver’s stress detection models for the low, medium, and high-stress classes, classification metrics are assumed which are defined in section III-E. EDAS is a fuzzy logic approach which is explained in section III-F is used here to provide a comprehensive ranking estimation of the proposed models for the three stress classes.

**A. PERFORMANCE EVALUATION**

Summary of the seven proposed ECG-based driver’s stress detection models for three stress classes is listed in Table 1. These results are based on the training images acquired using different augmentation operations. The training graphs for the seven pre-trained networks are shown in Figures 6-12. Results show that Model 5 based on Xception outperforms the other models based on GoogleNet, DarkNet-53, ResNet-101, InceptionResNetV2, DenseNet-201, InceptionV3 significantly by 11.32%, 11.32%, 9.45%, 7.54%, 5.66%, and 1.88% respectively with an overall validation accuracy of 98.11%. Confusion matrices for the seven pre-trained networks are shown in Figures 13-14. The performance of the proposed models for the three stress classes is shown in Figure 14(a) indicate the total number of correct predictions. In the first column, 23 HIGH instances are predicted correctly, while 0 and 1 HIGH instances are mispredicted as MEDIUM and LOW by the model. Thus, the total correct prediction for the HIGH-stress class is 95.8%. Similarly, for MEDIUM and LOW-stress classes, 10 out of 10 and 19 out of 19 instances were correctly predicted, which amounts to a total accuracy of 100% for both classes. Training results based on un-augmented images showed lower performance than augmented images, and some models also suffered from overfitting problems. To further investigate the performance of the proposed models for the three stress levels, the EDAS
method is used in the following subsection to provide a comprehensive ranking estimation.

**B. RANK-BASED PERFORMANCE EVALUATION**

EDAS method explained in section III-F is used here to provide a comprehensive ranking estimation of the proposed models for the three stress classes. The cross-efficient values ($\psi_\beta$) of various parameters of each of the proposed models are calculated for a particular driver’s stress level using Eqs. (9) and (10). Then the average positive and negative distances ($P_I$ and $N_I$) are calculated for each parameter of the models using Eqs. (13) and (16) respectively. The aggregate values ($SP_3$ and $SN_3$) are calculated using Eqs. (17) and (18). The aggregate values ($SP_3$ and $SN_3$) are then normalized using
Eqs. (19) and (20). Finally, the appraisal scores are measured using normalized aggregate values of \( N(SP_3) \) and \( N(SN_3) \) for ranking estimation of the proposed models using Eq. (21).

1) MEASUREMENTS FOR DRIVE’S HIGH-STRESS LEVEL

The performance metrics in Table 1 are considered as criteria for the pre-trained networks to classify the driver’s high-stress level. To estimate the ranking of different pre-trained
Networks, EDAS follows eight steps. In step 1, the aggregate calculation of Eqs. (9) and (10) can be obtained as the average cross-efficient value ($\psi$) for every criterion calculated values against each given in Table 2. In step 2 and 3, average positive and negative distances ($P_I$, $N_I$) are calculated for each parameter of the models using Eqs. (13) and (16) respectively as shown in Table 3 and 4. In step 4 and 5, the aggregate values of ($P_2$) and ($N_2$) are calculated using Eqs. (17) and
(18) respectively as shown in Table 5 and 6. In step 6, the normalized aggregate values of \(P_3\) and \(N_3\) are calculated using Eqs. (19) and (20) respectively as shown in Table 7. In step 7, the appraisal scores \(\lambda\) is measured using normalized aggregate values of \(P_3\) and \(P_3\) using Eq. (21) as shown in Table 7. In the final step 8, the appraisal scores \(\lambda\) are determined in decreasing order and then perform the ranking estimation of the proposed models for the driver’s high-stress level. The best ranking model is the one having the lowest appraisal scores \(\lambda\). The final ranked results are shown in Table 7. It is apparent from the table that Model 5 based on Xception pre-trained network has the lowest appraisal scores \(\lambda\) so it is the best model for driver’s high-stress level detection.

2) MEASUREMENTS FOR DRIVE’s MEDIUM STRESS LEVEL
Using the previous procedure (i.e., Eqs. (9) to (21) and Tables 2 to 7, the appraisal scores \(\lambda\) are calculated for different models based on the driver’s medium stress level. The resulting ranking values are shown in Table 8 below. It is apparent from the table that Model 5 based on Xception pre-trained network has the lowest appraisal scores \(\lambda\), so it is the best model for driver’s medium stress level detection.

3) MEASUREMENTS FOR DRIVE’s LOW-STRESS LEVEL
Using the previous procedure (i.e., Eqs. (9) to (21) and Tables 2 to 7), the appraisal scores \(\lambda\) are calculated for different models based on the driver’s low-stress level. The resulting ranking values are shown in Table 9 below. It is apparent from the table that Model 4, based on the DenseNet-201 pre-trained network, has the lowest appraisal scores \(\lambda\), so it is the best model for driver’s medium stress level detection.

V. DISCUSSION
The aim of the proposed models was the ECG-based real-world driver’s stress levels classification using deep transfer learning approaches. The overall performance evaluation of the seven proposed models summarized in Table 1 shows that Model 5 based on Xception outperforms the other models based on GoogLeNet, DarkNet-53, ResNet-101, InceptionResNetV2, DenseNet-201, InceptionV3 significantly by 11.32%, 11.32%, 9.45%, 7.54%, 5.66%, and 1.88% respectively with a validation accuracy of 98.11%. The EDAS based rank estimation of the low, medium, and high-stress levels is Xception based Model 5, while the best model for low-stress level detection is DenseNet-201 based Model 4, as shown in Table 7, 8, and 9.

Performance comparison of the best achieved results from the proposed ECG-based deep transfer learning models with other existing stress detection models is shown in Table 10 based on accuracy, sensitivity, precision, F-score, and specificity. Healey and Picard [16], Zhang et al. [20], Chen et al. [4], Haouij et al. [21], Lopez-Martinez et al. [22], Vargas-Lopez et al. [23], Dalmeida and Masala [24], Lee et al. [17], Bianco et al. [25], Zontone et al. [26], Sun et al. [63], de Vries et al. [64],
Al Abdi et al. [65], and Can et al. [66] proposed different stress detection models. All these models are based on handcrafted feature extraction techniques, and different traditional machine learning algorithms were used to classify different stress levels. Although handcrafted features produce encouraging results, extracting the best features using these approaches is always time consuming and challenging task. Chen et al. [4], Healey and Picard [16], Haouij et al. [21], Lee et al. [17], Bianco et al. [25], Sun et al. [63], and Can et al. [66] proposed feature level fusion models which are inadequate for capturing the nonlinear correlation across different signals which appear simultaneously [32]. Moreover, the use of multiple physiological sensors may be unfeasible for constant stress monitoring due to system suitability issues during real-world driving. Lee et al. [17], Zhang et al. [20], Sun et al. [63], and Can et al. [66] have used Fast Fourier Transform (FFT) based techniques...
for features extraction which are not suitable for analyzing the non-stationary signals [28]. Lee et al. [17], Bianco et al. [25], and Zontone et al. [26] proposed driver’s stress detection models for simulated driving scenarios but the real-world driving situations are quite different from simulated driving conditions. Vargas-Lopez et al. [23], Dalmeida and Masala [24], Lee et al. [17], Bianco et al. [25], Zontone et al. [26], Sun et al. [63], de Vries et al. [64], and Al Abdi et al. [65] presented stress detection schemes which are based on only two stress classes. Lee et al. [17] has proposed a driver’s stress detection model and achieved an accuracy of 95.38% but the study is conducted in simulated driving scenarios using traditional machine learning based fusion models with only two stress classes. Lee et al. [37] have achieved an accuracy of 95.67% using deep learning based driver’s stress detection approach but used only two stress classes.
TABLE 1. Performance metrics of various pre-trained networks.

| Driver’s Stress Detection Model | Pretrained Network Used | Driver’s Stress Level | Performance Measure      |
|--------------------------------|-------------------------|-----------------------|--------------------------|
|                                |                         | High                  | ACCY: 0.8679, SSY: 0.9565, PRC: 0.7857, FS: 0.8628, SPCY: 0.8000 | Overall Accuracy: 86.79% |
|                                | GoogLeNet               | Medium                | ACCY: 0.9057, SSY: 0.6000, PRC: 0.8571, FS: 0.7059, SPCY: 0.9767 | Overall Accuracy: 96.23% |
|                                |                         | Low                   | ACCY: 0.9623, SSY: 0.9000, PRC: 0.9474, SPCY: 1 | Overall Accuracy: 96.67% |
|                                | InceptionV3             | High                  | ACCY: 0.9623, SSY: 0.9565, PRC: 0.9565, SPCY: 0.9667 | Overall Accuracy: 96.23% |
|                                |                         | Medium                | ACCY: 0.9623, SSY: 0.9000, PRC: 0.9000, SPCY: 0.9767 | Overall Accuracy: 96.23% |
|                                |                         | Low                   | ACCY: 1, SSY: 1, PRC: 1, SPCY: 1 | Overall Accuracy: 96.23% |
|                                | ResNet-101              | High                  | ACCY: 0.9057, SSY: 0.8697, PRC: 0.9091, SPCY: 0.8889 | Overall Accuracy: 93.33% |
|                                |                         | Medium                | ACCY: 0.8868, SSY: 0.7000, PRC: 0.7000, SPCY: 0.9302 | Overall Accuracy: 88.66% |
|                                |                         | Low                   | ACCY: 0.9811, SSY: 0.9524, PRC: 0.9756, SPCY: 0.9697 | Overall Accuracy: 96.67% |
|                                | DenseNet-201            | High                  | ACCY: 0.9245, SSY: 0.8261, PRC: 1, SPCY: 1 | Overall Accuracy: 98.11% |
|                                |                         | Medium                | ACCY: 0.9434, SSY: 1, PRC: 0.7692, SPCY: 0.9302 | Overall Accuracy: 92.45% |
|                                |                         | Low                   | ACCY: 0.9811, SSY: 0.9523, PRC: 0.9756, SPCY: 0.9697 | Overall Accuracy: 96.67% |
|                                | Xception                | High                  | ACCY: 0.9811, SSY: 0.9583, PRC: 0.9787, SPCY: 0.9667 | Overall Accuracy: 98.11% |
|                                |                         | Medium                | ACCY: 0.9811, SSY: 1, PRC: 1, SPCY: 1 | Overall Accuracy: 98.11% |
|                                |                         | Low                   | ACCY: 0.9811, SSY: 0.9500, PRC: 0.9744, SPCY: 1 | Overall Accuracy: 98.11% |
|                                | InceptionResNetV2       | High                  | ACCY: 0.9623, SSY: 0.9565, PRC: 0.9565, SPCY: 0.9667 | Overall Accuracy: 90.57% |
|                                |                         | Medium                | ACCY: 0.9434, SSY: 0.7000, PRC: 1, SPCY: 1 | Overall Accuracy: 90.57% |
|                                |                         | Low                   | ACCY: 0.9057, SSY: 0.8261, PRC: 0.8235, SPCY: 0.8788 | Overall Accuracy: 90.57% |
|                                | DarkNet-53              | Medium                | ACCY: 0.8679, SSY: 0.5882, PRC: 0.7404, SPCY: 0.8372 | Overall Accuracy: 86.79% |
|                                |                         | Low                   | ACCY: 0.9623, SSY: 0.9000, PRC: 0.9474, SPCY: 1 | Overall Accuracy: 86.79% |

TABLE 2. Cross-efficient values.

| Proposed Model | Pretrained Network Used | Performance Measure for Driver’s High Stress Level |
|----------------|-------------------------|---------------------------------------------------|
| Model 1        | GoogLeNet               | Accuracy: 0.8679, Sensitivity: 0.9565, Precision: 0.7857, F-Score: 0.8628, Specificity: 0.8000 |
| Model 2        | InceptionV3             | Accuracy: 0.9623, Sensitivity: 0.9565, Precision: 0.9565, F-Score: 0.9667, Specificity: 0.9667 |
| Model 3        | ResNet-101              | Accuracy: 0.9057, Sensitivity: 0.8697, Precision: 0.9091, F-Score: 0.8889, Specificity: 0.9333 |
| Model 4        | DenseNet-201            | Accuracy: 0.9245, Sensitivity: 0.8261, Precision: 1.0000, F-Score: 0.9048, Specificity: 1.0000 |
| Model 5        | Xception                | Accuracy: 0.9811, Sensitivity: 0.9583, Precision: 0.9787, F-Score: 0.9667, Specificity: 1.0000 |
| Model 6        | InceptionResNetV2       | Accuracy: 0.9623, Sensitivity: 0.9565, Precision: 0.9565, F-Score: 0.9667, Specificity: 0.9667 |
| Model 7        | DarkNet-53              | Accuracy: 0.9057, Sensitivity: 0.7826, Precision: 1.0000, F-Score: 0.8781, Specificity: 1.0000 |

classes and multiple physiological signals. Rastgoo et al. proposed hybrid deep learning based driver’s stress detection model for three stress classes using multimodal data. The model achieved an accuracy of 92.8% during simulated driving conditions.

Table 10 shows that the proposed Model 5 has achieved the highest 98.11% accuracy based on a single physiological signal and three stress levels compared to the existing traditional machine learning and deep learning approaches. The overall results suggest that deep pre-trained network
### TABLE 3. Analysis results of average ($P_3$).

| Proposed Model | Pretrained Network Used | $P_3$ Measurements for Driver’s High Stress Level |
|----------------|-------------------------|--------------------------------------------------|
|                |                         | Accuracy | Sensitivity | Precision | F-Score | Specificity |
| Model 1        | GoogLeNet               | 0.0667   | 0           | 0.1623    | 0.0601  | 0.1557      |
| Model 2        | InceptionV3             | 0        | 0           | 0         | 0       | 0           |
| Model 3        | ResNet-101              | 0.0588   | 0.0907      | 0.0495    | 0.0706  | 0.0345      |
| Model 4        | DenseNet-201            | 0        | 0           | 0         | 0       | 0           |
| Model 5        | Xception                | 0        | 0           | 0         | 0       | 0           |
| Model 6        | InceptionResNetV2       | 0        | 0           | 0         | 0       | 0           |
| Model 7        | DarkNet-53              | 0.0260   | 0.1370      | 0         | 0.0435  | 0           |

### TABLE 4. Analysis results of average ($N_3$).

| Proposed Model | Pretrained Network Used | $N_3$ Measurements for Driver’s High Stress Level |
|----------------|-------------------------|--------------------------------------------------|
|                |                         | Accuracy | Sensitivity | Precision | F-Score | Specificity |
| Model 1        | GoogLeNet               | 0        | 0.0547      | 0         | 0       | 0           |
| Model 2        | InceptionV3             | 0.0348   | 0.0547      | 0.0197    | 0.0418  | 0.0201      |
| Model 3        | ResNet-101              | 0        | 0           | 0         | 0       | 0           |
| Model 4        | DenseNet-201            | 0        | 0           | 0.0454    | 0       | 0.0344      |
| Model 5        | Xception                | 0.0832   | 0.2777      | 0         | 0.1145  | 0           |
| Model 6        | InceptionResNetV2       | 0.0348   | 0.0547      | 0.0197    | 0.0418  | 0.0201      |
| Model 7        | DarkNet-53              | 0        | 0           | 0.0660    | 0       | 0.0552      |

### TABLE 5. Analysis results of the aggregate ($P_3$).

| Proposed Model | Pretrained Network Used | Aggregate $P_3$ Measurements for Driver’s High Stress Level $(SP_3)_\alpha$ |
|----------------|-------------------------|-------------------------------------------------------------------------|
|                |                         | Accuracy | Sensitivity | Precision | F-Score | Specificity |
| Model 1        | GoogLeNet               | 0.0278   | 0           | 0.0235    | 0.0051  | 0.0105      | 0.06707 |
| Model 2        | InceptionV3             | 0        | 0           | 0         | 0       | 0           | 0       |
| Model 3        | ResNet-101              | 0.0108   | 0.0116      | 0.0044    | 0.00268 | 0.0011      | 0.0307 |
| Model 4        | DenseNet-201            | 0.0025   | 0           | 0         | 0.0045  | 0           | 0.007   |
| Model 5        | Xception                | 0        | 0           | 0.0060    | 0       | 0.0023      | 0.0084 |
| Model 6        | InceptionResNetV2       | 0        | 0           | 0         | 0       | 0           | 0       |
| Model 7        | DarkNet-53              | 0.0108   | 0.0390      | 0         | 0.0036  | 0           | 0.0536 |

### TABLE 6. Analysis results of the aggregate ($N_3$).

| Proposed Model | Pretrained Network Used | Aggregate $N_3$ Measurements for Driver’s High Stress Level $(SN_3)_\alpha$ |
|----------------|-------------------------|-------------------------------------------------------------------------|
|                |                         | Accuracy | Sensitivity | Precision | F-Score | Specificity |
| Model 1        | GoogLeNet               | 0        | 0.0156      | 0         | 0       | 0           | 0.0156 |
| Model 2        | InceptionV3             | 0.0145   | 0.0156      | 0.0028    | 0.0035  | 0.0013      | 0.0379 |
| Model 3        | ResNet-101              | 0        | 0           | 0         | 0       | 0           | 0       |
| Model 4        | DenseNet-201            | 0        | 0           | 0.0066    | 0       | 0.0023      | 0.0089 |
| Model 5        | Xception                | 0.0347   | 0.0791      | 0         | 0.0096  | 0           | 0.1236 |
| Model 6        | InceptionResNetV2       | 0.0145   | 0.0156      | 0.0028    | 0.0035  | 0.0013      | 0.0379 |
| Model 7        | DarkNet-53              | 0        | 0           | 0.0096    | 0       | 0.0037      | 0.0133 |
models are suitable for stress modeling independent of any feature engineering as such models can automatically recognize the high level of data abstraction. Although the obtained results are hopeful regarding the relevance and efficiency of deep transfer learning for stress modeling; however, several parameters require tuning to obtain the optimal performance. The proposed models are based on CWT and CNN-based transfer learning techniques. The scalogram images obtained using the high-resolution CWT analysis of the driver’s ECG signals contain the most relevant components regarding the three stress levels. Deep transfer learning techniques construct reliable features automatically, classify the stress levels with high accuracy, and reduce the computational cost and time. Moreover, the proposed models are based on only ECG signals as compared to other schemes which have used the fusion of two or more physiological signals which may not be suitable for continuous stress surveillance during real-world driving.

VI. CONCLUSION AND FUTURE WORK

In this paper, real-world driver’s stress models using different pre-trained networks were proposed. Specifically, seven pre-trained networks (GoogLeNet, DarkNet-53, ResNet-101, InceptionResNetV2, Xception, DenseNet-201, and InceptionV3) were used to extract features from ECG based scalogram images to improve the detection performance automatically. Previous work in this area is heavily based on traditional machine learning models, which strongly rely on feature engineering. Extracting the best features using these approaches is always a challenging task. Recently, CNN-based deep learning techniques have been extensively applied for stress modeling to provide embedded automatic
feature extraction mechanisms. CNN provides gradual connections among different layers, thus causing either gradient exploding or vanishing problems when expended beyond a certain limit. Moreover, acquiring a large dataset for training an entire CNN from scratch is also a challenging task.

The performance of the proposed driver’s stress detection models was assessed on ECG signals acquired from PhysioNet real-world driving database. Results showed that deep transfer learning approaches increased the accuracy levels of driver’s stress models compared to the traditional machine learning approaches.
learning and deep learning approaches. The reason is that the pre-trained networks already trained on a large benchmark dataset can perform automatic features extraction and training for a small target dataset efficiently without over-fitting problems. The current work is based on only ECG signals for stress modeling, so other signals from the same database should be included in future work. Moreover, the current approach needs to be examined on other datasets with multimodal data comprising physiological signals and other information regarding the driver, vehicle, and environment.

REFERENCES

[1] G. W. Evans and S. Carrère, “Traffic congestion, perceived control, and psychophysiological stress among urban bus drivers,” J. Appl. Psychol., vol. 76, no. 5, pp. 658–663, Oct. 1991.

[2] I. Hanzlíková, “Professional drivers: The sources of occupational stress,” in Prenauw za Prezentaion do Seminario Young Researchers Seminar, 2005. [Online]. Available: http://www.cctri.org/YRS05/Papers/Session-4/hanzlikova.pdf

[3] C. J. D. Naurois, C. Bourdin, A. Stratulat, E. Diaz, and J. L. Vécher, “Detection and prediction of driver drowsiness using artificial neural network models,” Accident Anal. Prevention, vol. 126, pp. 95–104, May 2019.

[4] L.-L. Chen, Y. Zhao, P.-F. Ye, J. Zhang, and J.-Z. Zou, “Detecting driving stress in physiological signals based on multimodal feature analysis and kernel classifiers,” Expert Syst. Appl., vol. 85, pp. 279–291, Nov. 2017.

[5] M. Manser and A. Rienert, “Evaluation of driver stress while transiting road tunnels,” in Proc. Int. Conf. Automot. User Interfaces Interact. Veh. Appl., pp. 1–6.

[6] N. Benlagha and L. Charfeddine, “Risk factors of road accident severity and the development of a new system for prevention: New insights from China,” Accident Anal. Prevention, vol. 136, Mar. 2020, Art. no. 105411.

[7] I. M. R. Sapolsky, “Stress and the aging brain,” in Proc. 13th EAI Int. Conf. Pervasive Comput. Technol. Healthcare, vol. 126, pp. 1–6, Apr. 2021.

[8] S. Bianco, P. Napolitano, and R. Schettini, “Multimodal car driver stress recognition,” in Proc. 12th Int. Conf. Pervasive Comput. Technol. Proc. Int. Conf. Automot. User Interfaces Interact. Veh. Appl., vol. 1, pp. 1–6.

[9] G. W. Evans and S. Carrère, “Traffic congestion, perceived control, and psychophysiological stress among urban bus drivers,” J. Appl. Psychol., vol. 76, no. 5, pp. 658–663, Oct. 1991.

[10] I. Hanzlíková, “Professional drivers: The sources of occupational stress,” in Prenauw za Prezentaion do Seminario Young Researchers Seminar, 2005. [Online]. Available: http://www.cctri.org/YRS05/Papers/Session-4/hanzlikova.pdf

[11] C. J. D. Naurois, C. Bourdin, A. Stratulat, E. Diaz, and J. L. Vécher, “Detection and prediction of driver drowsiness using artificial neural network models,” Accident Anal. Prevention, vol. 126, pp. 95–104, May 2019.

[12] L.-L. Chen, Y. Zhao, P.-F. Ye, J. Zhang, and J.-Z. Zou, “Detecting driving stress in physiological signals based on multimodal feature analysis and kernel classifiers,” Expert Syst. Appl., vol. 85, pp. 279–291, Nov. 2017.

[13] M. Manser and A. Rienert, “Evaluation of driver stress while transiting road tunnels,” in Proc. Int. Conf. Automot. User Interfaces Interact. Veh. Appl., pp. 1–6.
Y. Xing, C. Lv, H. Wang, D. Cao, E. Velenis, and F. Y. Wang, “Driver activity recognition for intelligent vehicles: A deep learning approach,” IEEE Trans. Veh. Technol., vol. 68, no. 6, pp. 5379–5390, Jun. 2019.

K. Masood and M. A. Alghamdi, “Modeling mental stress using a deep learning framework,” IEEE Access, vol. 7, pp. 68446–68454, 2019.

S. Basar, M. Ali, G. Ochoa-Ruiz, A. Waheed, G. Rodriguez-Hernandez, and M. Zareei, “A novel defocused image segmentation method based on PCNN and LBP,” IEEE Access, vol. 9, pp. 87219–87240, 2021.

G. Mehmoond, M. Z. Khan, A. Waheed, M. Zareei, and E. M. Mohamed, “A trust-based energy-efficient and reliable communication scheme (trust-based ERCS) for remote patient monitoring in wireless body area networks,” IEEE Access, vol. 8, pp. 131397–131413, 2020.

S. Basar, M. Ali, G. Ochoa-Ruiz, M. Zareei, A. Waheed, and A. Adnan, “Unsupervised color image segmentation: A case of RGB histogram based K-means clustering initialization,” PLoS ONE, vol. 15, no. 10, Oct. 2020, Art. no. e0240015.

G. D. Clifford, F. Azuaje, and P. Mcsharry, “ECG statistics, noise, artifacts, and missing data,” Adv. Methods Tools ECG Data Anal., vol. 6, no. 12, p. 18, 2006.

L. Sörnmo and P. Laguna, Bioelectrical Signal Processing in Cardiac and Neurological Applications, vol. 8. New York, NY, USA: Academic, 2005.

C. Levkov, G. Mihov, R. Ivanov, I. Daskalov, I. Christov, and I. Dorsinsky, “Removal of power-line interference from the ECG: A review of the subtraction procedure,” Biomed. Eng. Online, vol. 4, no. 1, pp. 1–18, Dec. 2005.

D. Zhang, “Wavelet approach for ECG baseline wander correction and noise reduction,” in Proc. IEEE Eng. Med. Biol. 27th Annua. Conf., Jun. 2005, pp. 1212–1215.

A. Milchevski and M. Gussev, “Performance evaluation of FIM and IIR filtering of ECG signals,” in Advances in Intelligent Systems and Computing, vol. 665, 2018, pp. 103–112.

M.-F. Guo, X.-D. Zeng, D.-Y. Chen, and N.-C. Yang, “Deep-learning-based earth fault detection using continuous wavelet transform and convolutional neural network in resonant grounding distribution systems,” IEEE Sensors J., vol. 18, no. 3, pp. 1291–1300, Feb. 2018.

R. Wasem and Z. Wang, “Deep convolutional neural networks for image classification: A comprehensive review,” Neural Comput., vol. 29, no. 9, pp. 2352–2449, Aug. 2017.

J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, “How transferable are features in deep neural networks?” in Proc. Adv. Neural Inf. Process. Syst., vol. 4, Jan. 2014, pp. 3320–3328.

H.-C. Shin, H. R. Roth, M. Gao, L. Lu, Z. Xu, I. Nogues, J. Yao, D. Mollura, and R. M. Sammers, “Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning,” IEEE Trans. Med. Imag., vol. 35, no. 5, pp. 1285–1298, Feb. 2016.

G. Zeng, Y. He, Z. Yu, X. Yang, R. Yang, and L. Zhang, “Going deeper with convolutions Christian,” J. Chem. Technol. Biotechnol., vol. 91, no. 8, pp. 2322–2330, 2014.

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

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

K. He and J. Sun, “Convolutional neural networks at constrained time cost,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2015, pp. 5353–5360.

G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, “Densely connected convolutional networks,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jul. 2017, pp. 2261–2269.

F. Chollet, “Xception: Deep learning with depthwise separable convolutions,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jul. 2017, pp. 1800–1807.

C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, “Inception-v4, Inception-ResNet and the impact of residual connections on learning,” in Proc. AAAI Conf. Artif. Intell., 2017, pp. 4278–4284.

J. Redmon and A. Farhadi, “YOLOv3: An incremental improvement,” 2018, arXiv:1804.02767.

M. K. Ghorabae, E. K. Zavadskas, L. Olfat, and Z. Turskis, “Multi-criteria inventory classification using a new method of evaluation based on distance from average solution (EDAS),” Informatica, vol. 26, no. 3, pp. 435–451, Mar. 2015.

F.-T. Sun, C. Kuo, H.-T. Cheng, S. Buthpiitya, P. Collins, and M. Griss, “Activity-aware mental stress detection using physiological sensors,” in Proc. Int. Conf. Mobile Comput., Appl., Services, 2012, pp. 282–301.

J. J. G. de Vries, S. C. Pauws, and M. Bielh, “Insightful stress detection from physiology modalities using learning vector quantization,” Neurocomputing, vol. 151, pp. 873–882, Mar. 2015.

R. M. Al Abdi, A. E. Alhitary, E. W. Abdul Hay, and A. K. Al-bashir, “Objective detection of chronic stress using physiological parameters,” Med. Biol. Eng. Comput., vol. 56, no. 12, pp. 2273–2286, Dec. 2018.

Y. S. Can, N. Chalabianloo, D. Ekiz, and C. Ersoy, “Continuous stress detection using wearable sensors in real life: Algorithmic programming contest case study,” Sensors, vol. 19, no. 8, p. 1849, Apr. 2019.

MUHAMMAD ASIF received the B.Sc. degree in computer systems engineering from the KPK University of Engineering and Technology, Peshawar, Pakistan, in 2006. He received the Master of Science (M.S.) degree in electronics and communications engineering from Myongji University, South Korea, in 2009, and the Ph.D. degree in biomedical engineering from LISIN, Politecnico di Torino, in 2016, under the Erasmus Mundus Expert II Fellowship. He is currently acting as an Assistant Professor and the Head of the Software Engineering Department, University of Malakand. His research interests include MANETS, MWSN, and machine learning.

MUHAMMAD AMIN received the M.Sc. degree in electronics from the Department of Electronics, University of Peshawar, in 1999, and the M.S. degree in computer science from the Faculty of Management and Computer Sciences, University of Agriculture, Peshawar, in 2010. He is currently working as an Assistant Professor with the Department of Computer Science, Iqra National University, Pakistan. His main research interests include MANETS, MWSN, and machine learning.

KHALIL ULLAH graduated in computer systems engineering from the University of Engineering and Technology, Peshawar, Pakistan, in 2006. He received the Master of Science (M.S.) degree in electronics and communications engineering from Myongji University, South Korea, in 2009, and the Ph.D. degree in biomedical engineering from LISIN, Politecnico di Torino, in 2016, under the Erasmus Mundus Expert II Fellowship. He is currently acting as an Assistant Professor and the Head of the Software Engineering Department, University of Malakand. His research interests include extracting muscle anatomical and physiological information from high-density electromyography, computer vision, digital signal and image processing, and deep learning with applications to medical healthcare.
ABDUL WAHEED received the master’s and Ph.D. degrees in computer science from the Department of Information Technology, Hazara University Mansehra, in 2014 and 2021, respectively. He has completed his Ph.D. research at NetLab-INMC under the School of Electrical and Computer Engineering (ECE), Seoul National University (SNU), South Korea, in 2019, under the HEC Research Program. He is currently a member of the Crypto-Net Research Group, Hazara University. He also works as an Assistant Professor with the Department of Computer Sciences and the Dean of the Faculty of Engineering and Information Technology (FEIT), Northern University, Nowshera. He has numerous publications in journals and international conferences. His research interests include information security, secure and smart cryptography, heterogeneous communications within IoT, mobile ad-hoc networks (MANETs), wireless sensor networks (WSNs) security, and fuzzy logic-based decision-making theory.

SANA UL HAQ received the B.Sc. degree in electrical engineering from the University of Engineering and Technology, Peshawar, Pakistan, in 2002, and the Ph.D. degree in multimodal emotion classification from CVSSP, University of Surrey, U.K., in 2011. He worked as a Systems Engineer in the telecom industry, from 2003 to 2006. He is currently working as an Assistant Professor at the Department of Electronics, University of Peshawar, Pakistan. His research interests include human–computer interaction, pattern recognition, multimodal emotion recognition, and speech and image processing.

MAHDI ZAREEI (Senior Member, IEEE) received the M.Sc. degree in computer network from the University of Science, Malaysia, in 2011, and the Ph.D. degree from the Communication Systems and Networks Research Group, Malaysia-Japan International Institute of Technology, University of Technology, Malaysia, in 2016. In 2017, he joined the School of Engineering and Sciences, Tecnológico de Monterrey, as a Postdoctoral Fellow, where he has been working as a Research Professor, since 2019. His research interests mainly include wireless sensor and ad-hoc networks, energy harvesting sensors, information security, and machine learning. He is a member of the Mexican National Researchers System (Level I). He is serving as an Associate Editor for IEEE Access and Ad-Hoc and Sensor Wireless Networks.

R. R. BISWAL received the B.Sc. and M.Sc. degrees in solid state physics from Sambalpur University, India, the second master’s degree in nanotechnology from Amity University, India, and the Ph.D. degree in electrical engineering from CINVESTAV-IPN, Mexico. His postdoctoral research was carried out at IER-UNAM, Mexico. He joined the School of Engineering and Sciences,ITESM, Guadalajara Campus, as a Professor of mathematics and data science. He is currently a member of the ITESM Artificial Intelligence and Data Science Hub, where he participates as a Consultant. He is a Distinguished Researcher at the Mexican research evaluation system called “SNI” and collaborates with various industries, such as the regional citrus industry in Veracruz, the Bank of Mexico, restaurants in Veracruz, the Hospital de Mujeres de Alta Specialty of Tabasco, Jalisco Institute of Cancerology, and the Government of Jalisco for specific solutions that use machine learning and artificial intelligence. He is also the Director of Algorithmics, Guadalajara, a programming school for kids in Guadalajara. He is truly fascinated with the amazing world of data science and machine learning and aspires to greatly contribute to the emerging industries and future leaders of Mexico.

* * *