IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 21 October 2022, accepted 28 October 2022, date of publication 1 November 2022, date of current version 8 November 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3218711

RESEARCH ARTICLE

Design of an Efficient Distracted Driver Detection System: Deep Learning Approaches

NAVEEN KUMAR VAEGAE[1], (Senior Member, IEEE), KRANTHI KUMAR PULLURI[1], KALAPRAVEEN BAGADI[1], (Member, IEEE), AND OLUTAYO O OYERINDE[2], (Senior Member, IEEE)

[1]Vellore Institute of Technology, Vellore 632014, India
[2]University of the Witwatersrand, Johannesburg 2050, South Africa

Corresponding authors: Olutayo O Oyerinde (olutayo.oyerinde@wits.ac.za) and Kalapraveen Bagadi (kpbagadi@gmail.com)

This work was supported in part by the National Research Foundation (NRF) of South Africa through the NRF Competitive Program funding for Rated Researchers under Grant 118547 and the NRF, South Africa/Polish National Center for Research and Development (NCBR), Poland Joint Science, and Technology Research Collaboration under Grant 118678.

ABSTRACT Distracted driving is any activity that deviates an individual's attention from driving. Some of these activities include talking to people in the vehicle, using hand-held devices such as mobile phones or tablets, eating or drinking, and adjusting the stereo or navigation systems while driving. To counter the effects caused by distracted driving, many countries around the world have imposed rules and charged fines on distracted drivers in order to ensure safe driving. Owing to the technological advancement in recent times, modern-day technologies such as computer vision, image processing, and machine learning techniques can further support the efforts of governments to prevent accidents caused by distracted driving. In this paper, an efficient distracted driver detection scheme (DDDS) has been proposed using two robust deep learning architectures, mainly visual geometric groups (VGG-16) and residual networks (ResNet-50). The proposed DDDS scheme contains the pre-processing module, image augmentation techniques, and two classification modules based on deep learning architectures. Both the architectures are implemented, and the results have been compared in terms of performance indices, namely accuracy, and logarithmic loss. The two-dimensional (2D) dashboard images derived from the State-Farm dataset are pre-processed and are used for training, testing, and validation of the proposed architectures. Accuracy of 86.1% and 87.92% are achieved with VGG-16 and ResNet-50 models, respectively, and it is observed that the DDDS scheme is found highly efficient for c4, c5, and c7 categories of the State-Farm dataset. The results obtained with the proposed DDDS methodology are compared with existing literature and found to be satisfactory. The algorithms developed and discussed for the proposed DDDS can be instrumental in reducing the fatalities and injuries caused due to distracted driving.

INDEX TERMS Distracted driving, deep learning, ResNet-50, state-farm dataset, VGG-16.

I. INTRODUCTION

The motivation for this paper stems from the immense need to reduce the surge in the number of traffic accidents in the recent past. According to the World Health Organization (WHO), road accidents are said to be responsible for over 1.3 million deaths throughout the world every year and are often touted as a 'global tragedy' [1]. Vehicular accidents are on a constant rise every year. Among numerous factors responsible for road accidents, one of the foremost reasons for the occurrence of these accidents can be pinpointed as distracted driving by drivers. Statistics indicate that 84 people were injured per 100 in crashes involving distracted driving in 2019. In the United States, according to the National Highway Traffic Safety Administrator of the United States (NHTSA), over 23,744 vehicle occupants and 36,096 vehicle

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

non-occupants were killed in road accidents due to distracted driving in 2019 [2]. Thus, it is required to have an efficient distracted driver detection system to avoid such damage.

A majority of distractions on the road while driving has been credited to multitasking. Nowadays, multitasking has become an inclusive part of people's lives, especially teen drivers. It is proven that 21% of distracted driving accidents involve teens since smartphones and electronic gadgets are major sources of distraction [3]. A distracted span as short as 5 seconds can lead to a fatal accident [4]. Large vehicles like trucks are more prone to accidents than small vehicles. Statistics indicate that 71% of truck fatalities occur due to distracted driving [5]. Research also suggests that innocuous acts like reaching out for objects increase the chance of an instantaneous crash by eight times [6]. Distractions can occur for many reasons, such as manual, visual, or cognitive. The primary source of distractions are activities like eating, using GPS, applying makeup, combing hair, texting, reading maps, surfing the web, watching videos, etc. The percentage of distracted drivers by type of activity has been visualized in the bar graph shown in Fig. 1.

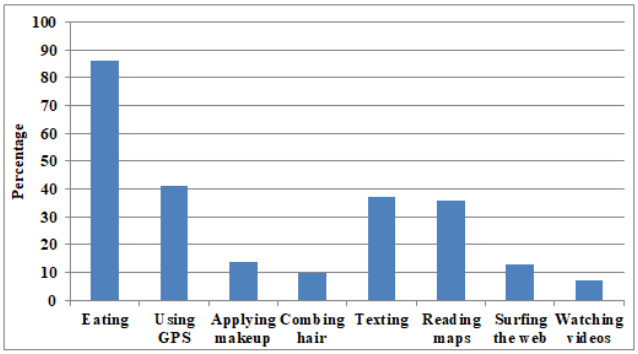


FIGURE 1. Percentage of distracted drivers by activity [7].

The paper is organized as follows. The works related to driver distraction are discussed in Section II. The workflow of the proposed DDDS is presented in section III. It mainly shows the State-Farm data set, pre-processing and image augmentation of data, and process of building the model. Section IV depicts the proposed neural network architectures. The results of the proposed scheme, along with performance evaluation, are illustrated in section V. Section VI concludes the work with important inferences.

II. RELATED WORKS

There have been numerous efforts to solve the problem of distracted driving in the past. K.Sengupta et al. [8] performed driver sleep detection using machine learning methods but didn't focus on other types of distractions. Since the most frequent distracted driving cases involved the usage of cell phones, researchers set out to detect cell phone usage while driving. Zhang et al. [9] created one of the first databases of several images captured through the camera mounted on the dashboard and used conditional random fields to detect the usage of mobile phones during driving. Many other researchers additionally used this database to detect texting during cell phone usage. The results conclude that usage of smartphones increase driving risks [10]. N.Das et al. [11] created and compiled a dataset based on hand movement detection in an automotive environment using aggregate channel features (ACF). H.N.Esfahani et al. [12] used a driving simulator to study the impact of reading and messaging while driving. K.Seshadri et al. [13] and T.H.N.Le et al. [14] created their own datasets and proposed an automated system for determining the coordination between the driver's cellphone usage and hands-on steering wheel detection. Researchers extensively used histogram of gradient (HoG) [15], AdaBoost [16], and hidden conditional random field model (HCRF) [15] to implement a distracted driver detection scheme (DDDS) on their self-collected datasets. It was only later that researchers resorted to using pre-trained models to extract the distracted driver features, as seen in D.Tran et al. [15]. Both support vector machine classifiers (SVM) [17] and convolutional neural networks (CNNs) [18] were adopted extensively on these pre-trained models. Y.Xing et al. [19] adopted a deep CNN model that used Kinect cameras to collect images to create a dataset. Self-made feature extractors like Scale Invariant Feature Transform (SIFT) and HoG combined with classical classifiers like CNNs and SVM were proposed as well. However, CNN's proved to be a better technique due to its efficiency and speed of computation [20]. C.Huang et al. [21] proposed a hybrid CNN framework (HCF) for recognizing the behavior of distracted drives. An accuracy of 96.4% was achieved using the HCF method.

F.Omerustaoglu et al. [22] integrated vision-based distracted driver detection models with sensor data to detect different driver distractions. Authors from real-world drivers considered nine distraction activities and generated a dataset made of sensor data and driver images. The model is built in two stages. The first stage consists of CNN models using transfer learning and fine-tuning methods. The second stage consists of LSTM-RNN models using fine-tuning methods. Accuracy of 62% and 76% is obtained with CNN and LSTM methods with only image data as an input. In contrast, an accuracy of 85% is obtained with both hybrid and prediction-level fusion LSTM methods by fusing the sensor data and image data as input. N.Mofid et al. [23], on the state farm distraction driving dataset, combined classification models with various data augmentations, OpenCV, and skin segmentation models to detect the type of driver distraction. The proposed Resnet-50 with random rotation and brightness train set augmentation, blur augmentation, and skin segmentation to achieve a F1-score of 0.662, which is 15% more than Resnet-50. K.Roy et al. [24], to detect driver distraction because of mobile usage while driving, designed an unsupervised learning method called a low rank sparse non-negative dictionary (LRSNND). With the LRSNND method, an accuracy of 77.19% was obtained with the state farm distraction driving dataset. H.V.Koay et al. [25] evaluated the efficiency of vision

transformers and CNN models in detecting distracted drivers. Author implemented 16 vision transformers models and 17 CNN models. The authors claim that CNN models performed well over the vision transformers models.

From the literature, it is visible that there is a demand for an efficient and robust model for the detection of driver distraction. This motivates us to propose an efficient distracted driver detection scheme (DDDS) using advanced deep learning approaches, namely VGG-16 and ResNet-50 neural network frameworks. The data obtained from the State-Farm dataset is subjected to data processing and image augmentation in the preprocessing stage. This scheme uses two-dimensional (2D) dashboard images derived from the State-Farm dataset to train, test, and validate the proposed architectures. Accuracy and logarithmic loss are the metrics that have been employed to gauge the performance of each model. The results obtained with the proposed DDDS methodology are compared with existing literature.

III. PROPOSED DISTRACTED DRIVER DETECTION SYSTEM

The block diagram of the proposed DDDS is shown in Fig. 2. The data obtained from the State-Farm dataset is categorized into training and testing subsets. In the first stage, the data from the State-Farm dataset is subjected to data pre-processing and image augmentation. The model for DDDS is built using two neural network architectures, namely the VGG-16 and ResNet-50. The results of the VGG-16 and ResNet-50 neural network frameworks have been compared in terms of accuracy and logarithmic loss performance indices. The following sections discuss each block of the proposed DDDS.

A. THE STATE-FARM DATASET

The data has been derived from the State-Farm dataset of 2D dashboard images which are taken inside a car while the driver is driving [26]. A snippet of the images is shown in Fig. 3. The dataset has been split into testing and training images as per the 80/20 rule, respectively. The training subset has been labeled in two ways, mainly as i) per the class it belongs to (c0-c9) and ii) as per the person it belongs to (p002-p081). The distribution of the overall number of sample images in the training set with respect to the classes is visualized in Fig. 4. As seen in Fig. 4, it can be concluded that the images belonging to each class are in equal proportion to each other.

When the images were split according to the 80/20 rule at random, and the CNN model was deployed on the training set; data leakage was witnessed since there were numerous images belonging to different classes of the same person, often taken from fixed heights and angles that varied minimally. This led to the over-fitting of the model. In order to overcome this problem, instead of splitting the images at random, additional care was taken to split the data with respect to classes and persons to that they belong. Ultimately, the training set had approximately 22.4 thousand labeled samples with equal distribution among all classes (c0-c9) and person identities (p002-p081). In addition to the training set, an additional testing/validation set of 79.7 thousand unlabeled samples was also formed.

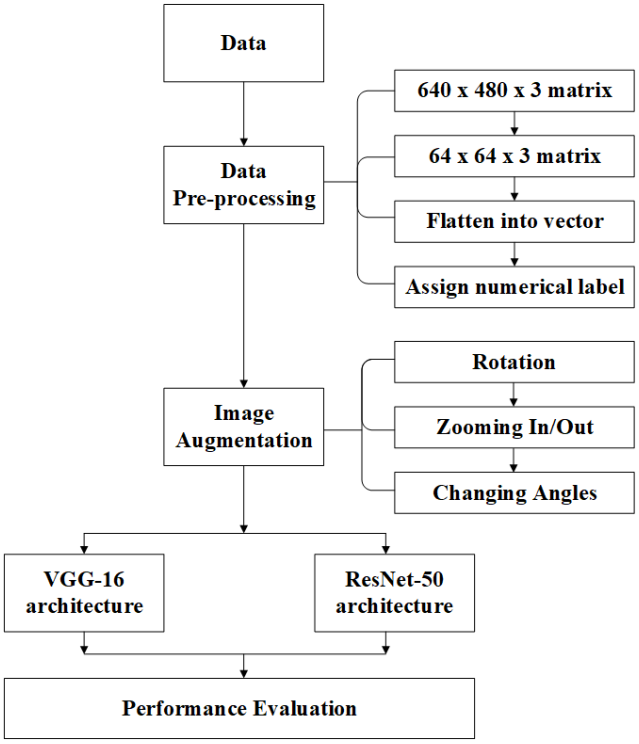


FIGURE 2. Block diagram of proposed DDDS model.

B. DATA PRE-PROCESSING

Each image is pre-processed before being passed to the classifier. Based on the RGB pixel values, each image is converted into a high-dimensional $640 \times 480 \times 3$ matrix, resized into a $64 \times 64 \times 3$ matrix using the Python cv2 library to improve the computing efficiency of the classifier, and finally flattened into a vector. A numerical label is assigned for each flattened vector according to the specified class.

C. IMAGE AUGMENTATION

Image augmentation is implemented on the pre-processed data. Image augmentation refers to the generation of additional sample images in the dataset through multiple and random stages of image manipulation, such as modification of the image dimensions, rotation, zooming in or out, altering the image angles, and so on [27]. Since the training set has multiple images of the same person taken from one fixed angle (due to the fixed position of the data collection apparatus, this case, a camera in the car), as seen in Fig. 5, the model might often confuse between the similar-angled pictures, which may lead to the reduction in the accuracy of the model. Thus, in these types of cases, image augmentation generates images from multiple angles, thereby resolving the problem to a great extent. Another important advantage of image augmentation is data multiplication. Greater the

FIGURE 3. A snippet of the State–Farm dataset and its classes (c0–c9).

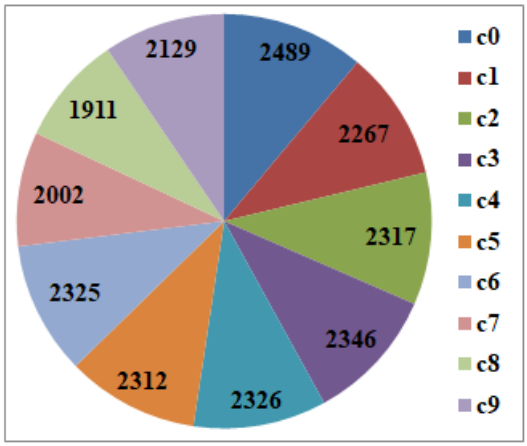


FIGURE 4. Number of training subset samples in each class (c0–c9).

number of images present to train the model, the greater accuracy of the model [28].

D. BUILDING THE MODEL

1) TRANSFER LEARNING AND THE IMAGENET CHALLENGE

The DDDS model is built using transfer learning, i.e., a machine learning process wherein a model trained to solve a particular task is applied and used as a starting point to build a new model for solving a related task. The ImageNet database was first developed as a part of a standard world-wide Computer Vision challenge called the ImageNet large-scale visual recognition challenge (ILSVRC) [29]. The model weights obtained in the ILSVRC challenge can be applied to any image processing datasets that use deep learning and computer vision. The DDDS model presented in this paper utilizes a model previously trained on the ImageNet database. However, the final Softmax layer of the pre-trained model is modified to suit the number of classes present in the State-Farm dataset. Hence, transfer learning has been adopted by using the ImageNet database weights as the model's starting point to detect distracted drivers on the State-Farm database.

(a) c2: talking on the phone

(b) c8: hair and makeup

FIGURE 5. Confusion between classes.

2) LAYER OF THE MODEL

Before delving into the DDDS model's specifications, a few layers are common to every model. These are described as follows:

Input Layer: The input layer is the first layer that the sample images (i.e., 224 × 224 RGB colour images) from the State-Farm datasets are passed through.

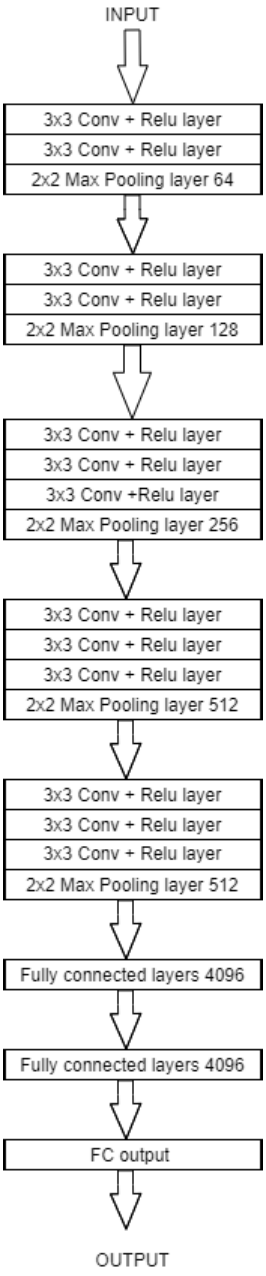


FIGURE 6. Diagrammatic representation of VGG–16 architecture.

Convolution Layer: It is a filter that utilizes a small matrix for which the number of rows and columns are signified. Thus, every image is passed through a stack of convolution layers, for which a receptive field of a standard 3 × 3 and a stride of 1 have been assigned. Every convolution kernel uses row and column padding; this is done to keep the size of the input and output features the same. This filter goes across the pixels of the image 3 × 3 at a time and, by convolving or sliding, gives a new image which is passed on to the next convolution layer.

Max Pooling/Max Stride: It is a layer used to reduce the dimensionality of the output images of CNN. Similarly, the

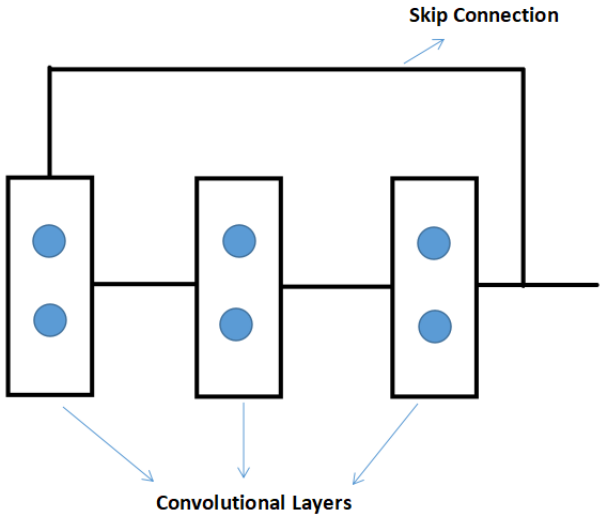


FIGURE 7. Skip connections.

Max Stride layer refers to the amount of sliding or convolving that needs to occur on the images.

In addition to the above-mentioned fundamental layers and the initial weights obtained from the ImageNet database, additional layers such as dense layers, batch normalization layers, global average pooling layers, and dropout layers are added to the DDDS model to maximize its performance and to maximize the outputs obtained after applying techniques of transfer learning on the pre-processed dataset.

IV. PROPOSED NEURAL NETWORK ARCHITECTURES

The VGG-16 and the ResNet-50 architectures have been used to develop the model for detecting distracted drivers. The work and methodology for each architecture have been explained in the following sub-sections.

A. VGG-16

The diagram of VGG-16 is depicted in Fig. 6. VGG-16 is one among the different implementations of the VGG, which has 16 layers of which 13 are convolutional layers. The 16 layers have Max pool and Soft-Max layers and even contain other layers that have trainable parameters. The ruling structure of any VGG model includes five sets of convolutional layers, after which a Max Pool layer follows up. The VGG-16 architecture starts with a low channel size of 64 and then eventually goes on to increase it by a factor of 2 after each max-pooling layer until it reaches 512. To sum up the architecture, it has two neighboring blocks of a couple of convolutional layers followed by a max-pooling, then it has three adjoining blocks of three convolutional layers, which is again followed by max-pooling, and at the end comes three dense layers. The last three convolutional layers have varied depths in different architectures. The quintessential aspect of being analyzed here is that the size is halved after each max-pooling. The rectified linear unit (RELU) layer refers to the rectifier unit,

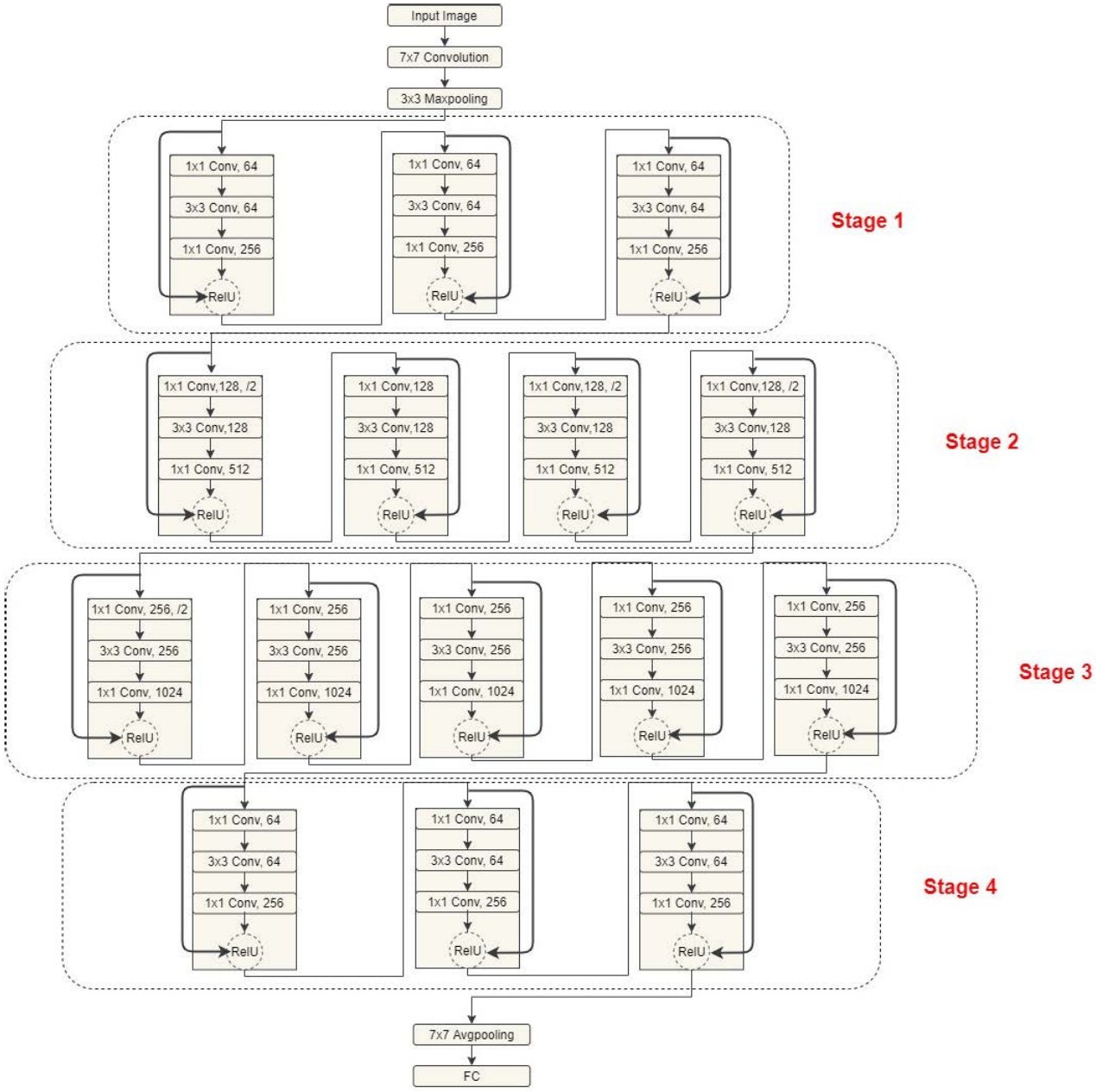


FIGURE 8. Diagrammatic representation of ResNet–50 architecture.

the most repeatedly used activation function for the outputs of the CNN neurons.

B. RESNET-50

A ResNet is a classical neural network that has been used extensively to solve tasks pertaining to computer vision. ResNets came to light during the ImageNet challenge in 2015. Before that, data scientists faced great difficulty training neural networks with more layers, mainly due to the ''vanishing gradient problem.'' As more layers are added to the neural network using activation functions, the gradients of the loss function approach zero, thus making the network hard to train. ResNets solved this problem by introducing the concept of skip connections, thus enabling data scientists to train neural networks with 150+ layers. Skip connections essentially provide residual connections between different stacked convolutional layers of the network. A residual/skip connection does not let the neural network pass through activation functions that are often responsible for minimizing the derivatives. Doing so results in the block achieving a higher overall derivative, thus making it easy for the neural network to be trained. Skip connections are illustrated in Fig. 7.

The ResNet-50 model consists of 4 stages, as shown in Fig. 8. In a nutshell, each stage has a convolutional block and an identity block. Furthermore, every convolutional block and identity block consists of 3 convolutional layers

each. The ResNet-50 is a highly efficient network with over 23 million trainable parameters. The input images of a ResNet-50 need to have their heights and widths in multiples of 32 and their channel widths as 3 for successful training. Once accepted by the network, the images are subjected to initial convolution through a 7×7 kernel size followed by the max-pooling layer of 3×3 kernel size. Max pooling is the operation used to select the maximum occurring element from a region of a feature map of the filter that often pinpoints the most prominent features of the image.

Stage 1 of the network consists of 3 residual blocks with three layers each. Each performs convolution using kernel sizes 64, 64, and 256, respectively. Similarly, stages 2, 3, and 4 have a varied number of residual blocks with three layers each, as seen in Fig. 8. Furthermore, skip connections, diagrammatically represented in Fig. 7, can be seen connecting the input layer directly to the output layer in the ResNet-50 model of Fig. 8. Moreover, each residual block has three convolutional layers performing 1 × 1, 3 × 3, and 1 × 1 convolutions respectively. The first 1 × 1 convolutional layer is responsible for reducing the dimensions of the input image. The 3 × 3 convolutional layers are a bottleneck layer for images with smaller dimensions. The second 1 × 1 convolutional layer restores the input image's dimensions. It is important to note that the channel width is doubled, and the size of the input image is reduced to half every time there is a progression from one stage to another. Lastly, the network uses an average pooling layer to select the average values of the pixels of the input image, followed by a fully connected layer with over 1000 neurons to give the output.

V. RESULTS

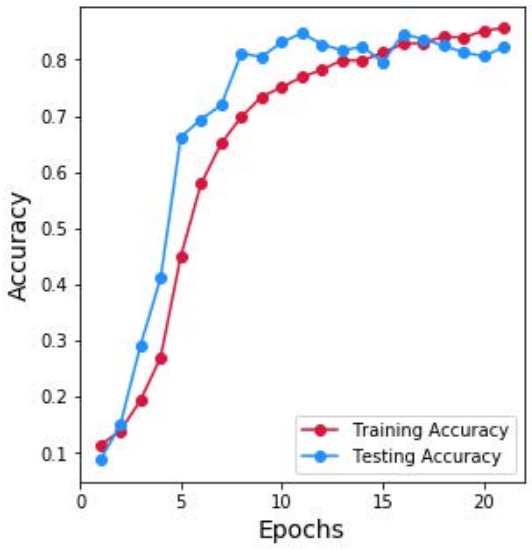
This section presents the results obtained by VGG-16 and ResNet-50 architectures. The performance of deep learning CNN models is evaluated using the numerical value of log loss functions. In this paper, both accuracies, as well as logarithmic loss, have been used as the metrics of performance. The models in this paper have been compared based on their train loss, test loss, and validation loss. The obtained results are compared with C.Huang et al. [21], F.Omerustaoglu et al. [22], and K.Roy et al. [24] which are available in the literature.

A. VGG-16

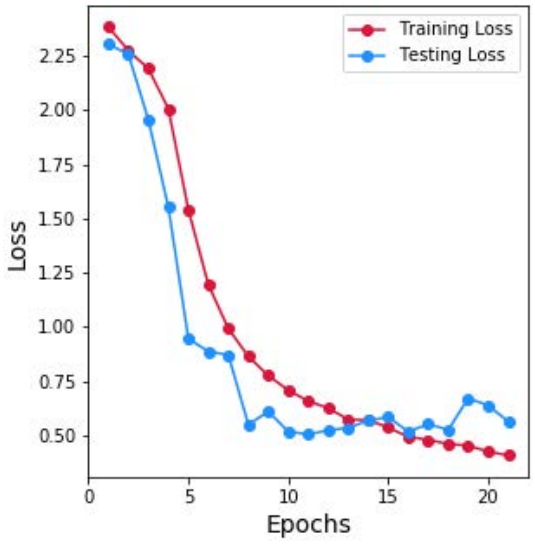
Upon applying the VGG-16 model to the dataset, an accuracy of 86.1 percent is achieved. Furthermore, the dropout accuracy graph in Fig. 9 clearly shows that when the dropout was increased beyond a certain threshold, the model could not fit properly.

B. RESNET-50

The ResNet-50 model achieved an accuracy of 87.92%, and the model was run for 25 epochs on the dataset to get the optimal value of both the accuracy and the logarithmic loss. The number of epochs was finalized when the accuracy did not improve and had reached the most minimal logarithmic loss value while the model was run for over 40 epochs. The visualization of accuracy and logarithmic loss value after each epoch has been shown in Fig. 10.



(a) Accuracy analysis



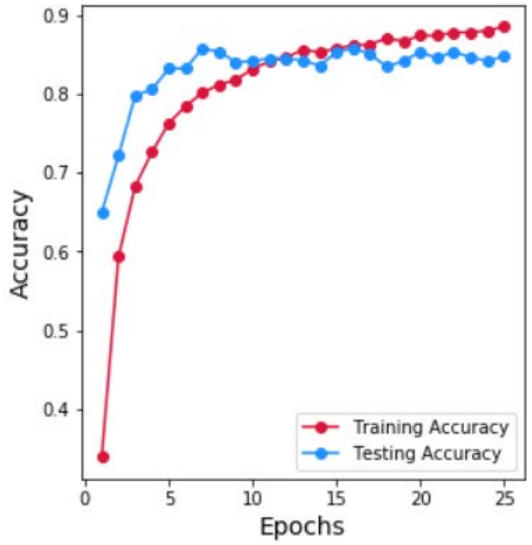
(b) Loss analysis

FIGURE 9. Performance metrics of results obtained by VGG-16 architecture.

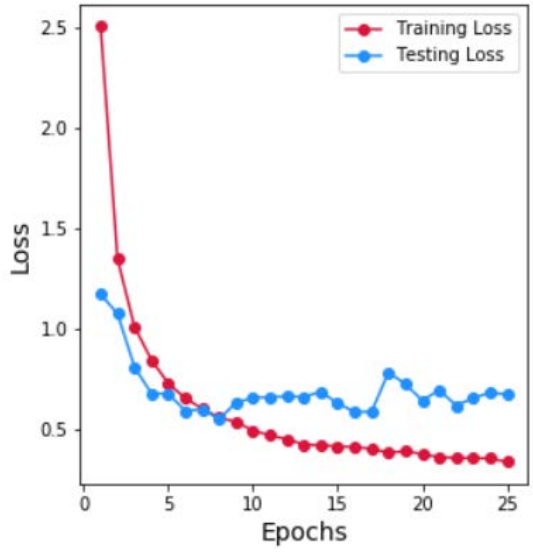
C. PERFORMANCE ANALYSIS

The summary of results produced on the State-Farm dataset by VGG-16 and ResNet-50 neural network architectures are shown in Table 1, wherein the correctly classified and misclassified images of both architectures have been presented. It can be seen that VGG-16 and ResNet-50 perform efficiently in categories c4, c5, and c7, respectively.

Moreover, the performance comparison of VGG-16 and ResNet-50 is presented in Table 2. The closer the proximity differences between the train and test losses, the higher the efficiency of the model. As seen in Table 2, the train and



(a) Accuracy analysis



(b) Loss analysis

FIGURE 10. Performance metrics of results obtained by ResNet-50 architecture.

test losses for VGG-16 are 0.41 and 0.64, respectively. This yields a difference of 0.23 between train and test losses. Similarly, the train and test losses for ResNet-50 are 0.34 and 0.57, respectively. This, yet again, yields a difference of 0.23 between train and test losses. Since the value of the difference is the same for both models, there cannot be a comparison between the efficiencies. Moving on to the validation loss, the values for VGG-16 and ResNet-50 are 0.5 and 0.55, respectively. Since minimal validation loss is preferable, VGG-16 performs well in this department. However, since the ResNet-50 model is fairly close to the validation loss of VGG-16, the most efficient model cannot be decided based on validation loss alone. Moving on to the accuracy, the VGG-16 and ResNet-50 models have obtained values of 86.1% and 87.92%, respectively. Since accuracy is a measure

TABLE 1. Sample results of correctly classified and misclassified images.

Categories	VGG-16		ResNet-50	
	Correctly classified images	Misclassified images	Correctly classified images	Misclassified images
c0	2441	289	2388	298
c1	2156	133	2188	204
c2	2285	183	2221	146
c3	2308	126	2339	122
c4	2310	73	2317	75
c5	2284	30	2294	34
c6	2171	165	2305	35
c7	1969	65	1950	158
c8	1784	190	1668	271
c9	1942	294	1939	291

of the correctness of predictions, ResNet-50 wins in this department. Yet again, VGG-16 has a fairly good accuracy value as well. Thus, it can be concluded that even though both VGG-16 and ResNet-50 are almost equal in performance, it comes down to the pros and cons of each model, as summarized in Table 2. The major factor that distinguishes both these architectures is the computational time. While the VGG-16 takes considerable time to process the data, the skip connections of the ResNet-50 do the same job relatively smoothly and, thus, allow the ResNet-50 model to yield results quickly. However, one major setback of the ResNet-50 model is its rapid accuracy saturation. Inspite of its high computational time, the VGG-16 trumps all its previous-generation models when it comes to performance. Hence, it can be said that deciding which neural network architecture to apply among VGG-16 and ResNet-50 purely depends on the nature of the problem statement and its outcomes.

D. COMPARISON

The results obtained with the proposed DDDS method are compared with the results of C. Huang et al. [21], F.Omerustaoglu et al. [22], and K.Roy et al. [24] in the next part of the sub-section.

TABLE 2. Performance comparision.

Parameter	VGG-16	ResNet-50
Train Loss	0.41	0.34
Test Loss	0.64	0.57
Validation Loss	0.5	0.55
Accuracy	86.1%	87.92%
Disadvantages	High computational time	Accuracy gets saturated rapidly
Advantages	Outperforms previous generation models	Residual connections reduce computational time

The results obtained using the DDDS methods are compared with the results of C. Huang et al. [21] and summarized in Table 3. C.Huang et al. designed HCF to recognize the behavior of distracted drives. The HCF model uses pre-trained CNN models for feature extraction and then concatenates the features to generate feature maps. Later fully connected layer is used to classify the distraction type. The accuracy in detecting driver behavior for categories c4, c5, c6, and c7 using our proposed DDDS method is higher by 0.33%, 1.23%, 1.07%, and 0.28%, respectively, than the HCF method designed by C.Huang et al.

TABLE 3. Comparison of proposed DDDS method with C.Huang et al. [21].

Categories	C.Huang et al. (2020) [21]	Proposed DDDS
c4	96.60%	96.93% (VGG-16)
c5	97.47%	98.70% (VGG-16)
c6	97.43%	98.50% (ResNet-50)
c7	96.52%	96.80% (VGG-16)

The next comparison is performed with F.Omerustaoglu et al. [22]. F.Omerustaoglu et al. [22] integrated vision-based distracted driver detection models with sensor data to detect different driver distractions. The detection model is designed in two stages, with the first stage consisting of CNN using transfer learning and fine-tuning methods. The second stage consists of LSTM-RNN models using fine-tuning methods. Results obtained with four methods by F.Omerustaoglu et al. [22] are compared with proposed DDDS methods and summarized in Table 4. Methods 1 and 2 are fine-tuning CNN and LSTM models with input as image data. Methods 3 and 4 are hybrid and prediction level fusion LSTM models with sensor and image data as input.

TABLE 4. Comparison of proposed DDDS method with F.Omerustaoglu et al. [22].

Techniques		Accuracy
F.Omerustaoglu et al. (2020) [22]	Method 1	62%
	Method 2	76%
	Method 3	85%
	Method 4	85%
Proposed DDDS Method	VGG-16	86.1%
	ResNet-50	87.92%

The last comparison is performed with K.Roy et al. [24] and summarized in Table 5. K.Roy et al. [24] detected mobile usage while driving using an unsupervised learning method called LRSNND. An accuracy of 77.19% is achieved with mobile usage classes of the state farm distraction driving dataset. With our proposed DDDS method considering mobile usage classes of state farm distraction driving dataset, the accuracy of 93.46% and 93.12% are achieved using VGG-16 and ResNet-50 respectively.

TABLE 5. Comparison of proposed DDDS method with K.Roy et al. [24].

Techniques		Accuracy
K.Roy et al. (2022) [24]	LRSNND	77.19%
Proposed DDDS Method	VGG-16	93.46%
	ResNet-50	93.12%

E. DISCUSSION

The main intention of this paper is to use the approach of image augmentation with VGG-16 and ResNet-50 neural network architectures to detect distracted drivers. Even though the proposed DDDS can provide a solution to distracted driving, similarities in images with respect to the postures lead to numerous misclassifications. However, this problem is solved by combining two or more architectures and including face-based approaches. Moreover, deep learning has also been used to enhance further accuracy [29]. After reviewing various phases of driver detection techniques and datasets, it can be concluded that VGG-16 and ResNet-50 best complement each other compared to the earlier works.

As a future enhancement, the proposed scheme can be part of a ubiquitous road safety system that can effectively monitor the driver's state and simultaneously actuate alerts and safety devices. Driver monitoring has become popular, and various automobile companies have started implementing such systems into their designs. By coupling internet of things (IoT) devices with the proposed model, warnings can be raised whenever the driver gets distracted by deploying the system on real-time data. Thus, it prevents any accidents due to the distractions of the driver. Moreover, this work can be advanced in the future by lowering the number of parameters and reducing computational time. Additionally, incorporating temporal context might help mitigate classification errors and increase the model's accuracy. Even though the deployment of machine learning schemes solves the visual and manual distractions problem, there must be an open challenge to work on cognitive distractions. This can be the contemporary enhancement of the proposed scheme to work.

VI. CONCLUSION

Distracted driving is a rising cause of concern today. This paper has elaborated upon using two neural network architectures, namely VGG-16 and ResNet-50, to identify distracted drivers. All the work was done on the State-Farm dataset with over lakh sample images. The data was first pre-processed and then augmented to obtain optimal results. A DDDS model was built through transfer learning and is deployed to classify the State-Farm images into one of the ten classes (c0-c9) of distracted driving. The architectures VGG-16 and ResNet-50 have effectively performed distracted driver detection with accuracies of 86.1% and 87.92% respectively. The proposed model is found to be efficient in the detection of c4, c5, and c7 categories of the State-Farm dataset. The results are compared with similar works available in the

existing literature. The performance comparison of the proposed schemes and similar works in the literature indicates deep learning schemes can efficiently and precisely detect distracted driving and circumvents accidents, reducing fatalities and injuries. The proposed DDDS can be a part of an intelligent dashboard of present-generation cars to monitor the driver's state effectively and simultaneously actuate alerts and safety devices.

ACKNOWLEDGMENT

The authors would like to thank the Vellore Institute of Technology, Vellore, India, and the University of the Witwatersrand, Johannesburg, South Africa, for supporting the collaborative research.

REFERENCES

[1] WHO. (Apr. 30, 2020). *Road Traffic Injuries-Key Facts*. Accessed: Aug. 1, 2022. [online]. Available: https://www.who.int/news-room/fact-sheets/detail/road-traffic-injuries
[2] U. S. Department of Transportation, "Traffic safety facts 2019—A compilation of motor vehicle crash data," Nat. Highway Traffic Saf. Admin., Washington, DC, USA, Tech. Rep. DOT HS 813 141, Aug. 2021. [Online]. Available: https://crashstats.nhtsa.dot.gov/Api/Public/ViewPublication/813141
[3] Edgar Snyder and Associates. *Texting and Driving Accident Statistics*. Accessed: Aug. 1, 2022. [online]. Available: https://www.edgarsnyder.com/car-accident/cause-of-accident/cell-phone/cell-phone-statistics.html
[4] Insurance Information Institute. *Facts + Statistics: Distracted Driving*. Accessed Aug. 1, 2022. [online]. Available: https://www.iii.org/fact-statistic/facts-statistics-distracted-driving/
[5] SMITH SYSTEM. *Smith System Announces—Driving Distracted*. Accessed: Aug. 1, 2022. [Online]. Available: https://www.prnewswire.com/news-releases/smith-system-announces-driving-distracted-300438049.html
[6] TEENSAFE. *100 Distracted Driving Facts and Statistics for 2018*. Accessed: Aug. 1, 2022. [Online]. Available: https://web.archive.org/web/20190118044311/https://www.teensafe.com/distracted-driving/100-distracted-driving-facts-and-statistics-2018/
[7] *Most U.S. Drivers Engage in 'Distracting' Behaviors: Poll*. Accessed: Aug. 24, 2022. [Online]. Available: https://consumer.healthday.com/mental-health-information-25/behavior-health-news-56/most-u-s-drivers-engage-in-distracting-behaviors-poll-659288.html
[8] K. Sengupta, A. Srivastava, S. Shreyansh, S. Aggarwal, and V. N. Kumar, "Driver sleep detection: A new and accurate approach," in *Proc. Innov. Power Adv. Comput. Technol. (i-PACT)*, 2021, pp. 1–7.
[9] X. Zhang, N. Zheng, F. Wang, and Y. He, "Visual recognition of driver hand-held cell phone use based on hidden CRF," in *Proc. IEEE Int. Conf. Veh. Electron. Saf.*, Jul. 2011, pp. 248–251.
[10] D. Beck and W. Park, "Perceived importance of automotive HUD information items: A study with experienced HUD users," *IEEE Access*, vol. 6, pp. 21901–21909, 2018.
[11] N. Das, E. Ohn-Bar, and M. M. Trivedi, "On performance evaluation of driver hand detection algorithms: Challenges, dataset, and metrics," in *Proc. IEEE 18th Int. Conf. Intell. Transp. Syst.*, Sep. 2015, pp. 2953–2958.
[12] H. Nasr Esfahani, R. Arvin, Z. Song, and N. N. Sze, "Prevalence of cell phone use while driving and its impact on driving performance, focusing on near-crash risk: A survey study in Tehran," *J. Transp. Saf. Secur.*, vol. 13, no. 9, pp. 957–977, Sep. 2021.
[13] K. Seshadri, F. Juefei-Xu, D. K. Pal, M. Savvides, and C. P. Thor, "Driver cell phone usage detection on strategic highway research program (SHRP2) face view videos," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2015, pp. 35–43.
[14] T. H. N. Le, Y. Zheng, C. Zhu, K. Luu, and M. Savvides, "Multiple scale faster-RCNN approach to driver's cell-phone usage and hands on steering wheel detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2016, pp. 46–53.
[15] D. Tran, H. Manh Do, W. Sheng, H. Bai, and G. Chowdhary, "Real-time detection of distracted driving based on deep learning," *IET Intell. Transp. Syst.*, vol. 12, no. 10, pp. 1210–1219, Dec. 2018.
[16] J. Hu, "Automated detection of driver fatigue based on AdaBoost classifier with EEG signals," *Frontiers Comput. Neurosci.*, vol. 11, p. 72, Aug. 2017.
[17] F. Tango and M. Botta, "Real-time detection system of driver distraction using machine learning," *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 2, pp. 894–905, Jun. 2013.
[18] H. M. Eraqi, Y. Abouelnaga, M. H. Saad, and M. N. Moustafa, "Driver distraction identification with an ensemble of convolutional neural networks," *J. Adv. Transp.*, vol. 2019, pp. 1–12, Feb. 2019.
[19] 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.
[20] M. D. Hssayeni, S. Saxena, R. Ptucha, and A. Savakis, "Distracted driver detection: Deep learning vs handcrafted features," *Electron. Imag.*, vol. 29, no. 10, pp. 20–26, Jan. 2017.
[21] C. Huang, X. Wang, J. Cao, S. Wang, and Y. Zhang, "HCF: A hybrid CNN framework for behavior detection of distracted drivers," *IEEE Access*, vol. 8, pp. 109335–109349, 2020.
[22] F. Omerustaoglu, C. O. Sakar, and G. Kar, "Distracted driver detection by combining in-vehicle and image data using deep learning," *Appl. Soft Comput.*, vol. 96, Nov. 2020, Art. no. 106657.
[23] N. Mofid, J. Bayrooti, and S. Ravi, "Keep your AI-es on the road: Tackling distracted driver detection with convolutional neural networks and targeted data augmentation," 2020, *arXiv:2006.10955*.
[24] K. Roy, "Unsupervised sparse, nonnegative, low rank dictionary learning for detection of driver cell phone usage," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 10, pp. 18200–18209, Oct. 2022.
[25] H. V. Koay, J. H. Chuah, and C.-O. Chow, "Convolutional neural network or vision transformer? Benchmarking various machine learning models for distracted driver detection," in *Proc. IEEE Region 10 Conf. (TENCON)*, Dec. 2021, pp. 417–422.
[26] KAGGLE. *State Farm Distracted Driver Detection*. Accessed: Aug. 1, 2022. [online]. Available: https://www.kaggle.com/c/state-farm-distracted-driver-detection/
[27] *Image Augmentation for Deep Learning*. Accessed: Aug. 24, 2022. [Online]. Available: https://towardsdatascience.com/image-augmentation-for-deeplearning-histogram-equalization-a71387f609b2
[28] *Building Powerful Image Classification Models Using Very Little Data*. Accessed: Aug. 24, 2022. [Online]. Available: https://blog.keras.io/building-powerful-image-classification-models-using-very-little-data.html
[29] IMAGENET. *ImageNet Large Scale Visual Recognition Challenge (ILSVRC)*. Accessed Aug. 1, 2022 [online]. Available: https://image-net.org/challenges/LSVRC/

NAVEEN KUMAR VAEGAE (Senior Member, IEEE) received the B.Tech. degree in instrumentation engineering from Nagarjuna University, Guntur, India, in 2001, the M.Tech. degree in electronics engineering from Jawaharlal Nehru Technological University, Kakinada, India, in 2010, and the Ph.D. degree in electrical and electronics engineering from the Vellore Institute of Technology, Vellore, India.

He is currently working as an Associate Professor with the School of Electronics Engineering, Vellore Institute of Technology. He has authored over nine SCI, and over 15 Scopus-indexed journals and conferences. His research interests include sensors and signal conditioning, signal processing, soft computing, intelligent systems, electronic nose, genomic signal processing, embedded systems, and the IoT.

Dr. Kumar is a Lifetime Member of ISTE and a member of the International Association of Engineering. He has been a Faculty Advisor and a Coordinator of IEEE Signal Processing Society, VIT Student Chapter, since 2017.

KRANTHI KUMAR PULLURI received the B.Tech. degree in electronics and communication engineering from Jawaharlal Nehru Technological University, Hyderabad, India, in 2008, and the M.Tech. degree in automotive electronics from the Vellore Institute of Technology, Vellore, India, in 2010, where he is currently pursuing the Ph.D. degree in electronics and communication engineering.

His research interests include sensors and signal conditioning, electronic nose, machine learning, and embedded systems.

KALAPRAVEEN BAGADI (Member, IEEE) received the B.E. degree in electronics and communication engineering from Andhra University, India, in 2006, the M.Tech. degree in electronic systems and communication from the National Institute Technology, Rourkela, India, in 2009, and the Ph.D. degree in wireless communication from the Department of Electrical Engineering, National Institute of Technology, Rourkela. He is currently working as a Professor with the School of Electronics Engineering (SENSE), Vellore Institute of Technology (VIT) Vellore, India. He has published several research articles in various journals and over 50 research article in refereed international journals and conferences. He has finished six Ph.D. theses and currently guiding two Ph.D. scholars. He is also a reviewer of the journals like *wireless personal communications*, *IET communications*, and *Telecommunication Systems*. His work has been cited more than 500 times at Google Scholar. His research interests include SDMA, MIMO, OFDM, wireless communication, and artificial intelligence.

OLUTAYO O OYERINDE (Senior Member, IEEE) received the Ph.D. degree in electronic engineering from the University of KwaZulu-Natal, Durban, South Africa, in 2011. Since 2013, he has been with the School of Electrical and Information Engineering, University of the Witwatersrand, Johannesburg, South Africa, where he is currently an Associate Professor.

He is a National Research Foundation (NRF) Rated Scientist, a Registered Professional Engineer (Pr.Eng.) with the Engineering Council of South Africa (ECSA), a Registered Engineer (R.Eng.) with COREN, and a Corporate Member of NSE. His current research interests include wireless communications, 5G and beyond 5G technologies, cognitive radio networks, and signal processing techniques for wireless communication systems. He is an Associate Editor of IEEE Access and an Editorial Board Member of the *International Journal of Sensors, Wireless Communications and Control*.