IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received December 9, 2021, accepted December 21, 2021, date of publication December 23, 2021,
date of current version December 31, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3138137

An Efficient Deep Learning Framework for Distracted Driver Detection

FAIQA SAJID[1], ABDUL REHMAN JAVED[2], (Member, IEEE), ASMA BASHARAT[1], NATALIA KRYVINSKA[3], ADIL AFZAL[4], AND MUHAMMAD RIZWAN[1]

[1]Department of Computer Science, Kinnaird College for Women, Lahore 54000, Pakistan
[2]Department of Cyber Security, Air University, Islamabad 44000, Pakistan
[3]Information Systems Department, Faculty of Management, Comenius University in Bratislava, 82005 Bratislava, Slovakia
[4]Bioinformatics Research Laboratory (BRL), Al-Khawarizmi Institute of Computer Science (KICS), University of Engineering and Technology (UET), Lahore 39161, Pakistan

Corresponding authors: Abdul Rehman Javed (abdulrehman.cs@au.edu.pk) and Natalia Kryvinska (natalia.kryvinska@fm.uniba.sk)

This work was supported by the Faculty of Management of Comenius University in Bratislava, Slovakia.

ABSTRACT The number of road accidents has constantly been increasing recently around the world. As per the national highway traffic safety administration's investigation, 45% of vehicle crashes are done by a distracted driver right around each. We endeavor to build a precise and robust framework for distinguishing diverted drivers. The existing work of distracted driver detection is concerned with a limited set of distractions (mainly cell phone usage). This paper uses the first publicly accessible dataset that is the state farm distracted driver detection dataset, which contains eight classes: calling, texting, everyday driving, operating on radio, inactiveness, talking to a passenger, looking behind, and drinking performed by 26 subjects to prepare our proposed model. The transfer values of the pertained model EfficientNet are used, as it is the backbone of EfficientDet. In contrast, the EfficientDet model detects the objects involved in these distracting activities and the region of interest of the body parts from the images to make predictions strong and accomplish state-of-art results. Also, in the Efficientdet model, we implement five variants: Efficientdet (D0-D4) for detection purposes and compared the best Efficientdet version with Faster R-CNN and Yolo-V3. Experimental results show that the proposed approach outperforms earlier methods in the literature and conclude that EfficientDet-D3 is the best model for detecting distracted drivers as it achieves Mean Average Precision (MAP) of 99.16% with parameter setting: learning rate of $le-3$, 50 epoch, batch size of 4, and step size of 250, demonstrating that it can potentially help drivers maintain safe driving habits.

INDEX TERMS Distracted driver detection, deep learning, convolution neural network (CNN), computer vision, distracted behaviour, intelligent transportation system, EfficientNet, EfficientDet.

I. INTRODUCTION

Numerous accidents on the roads happen because of the distraction of the driver. The AAA Foundation for traffic security found that 6 out of 10 road accidents are because of distracted drivers [1]. According to World Health Organization (WHO), 1.2 million deaths are due to road accidents, and 45% of road accidents are due to distracted drivers. The number of worldwide deaths in traffic accidents is nearly the same as the number of deaths due to Hepatitis and HIV (1.3 million [2], and 1.1 million [3], respectively). The centers for Disease Control and Prevention (CDC) provides a more precise definition of distracted driving, categorizing this into three types: 1) visual-looking around and not concentrating visually on the road, 2) cognitive-looking at the road but not concentrating mentally on the road and 3) and manual-taking the driver's hands off the steering wheel [4].

Road accidents cause loss of property and sometimes life. An increase in the number of road accidents caused by distracted driving has been noticed. According to the National Highway Traffic Safety Administration (NHTSA), distracted driving refers to any action that can redirect consideration from driving, including (a) talking or texting on the phone, (b) eating and drinking, (c) talking to others in the vehicle, or (d) using radio, entertainment or navigation system. 3091000 people were wounded, 3,477 died in 2015, and distracted drivers caused car accidents. The use of cell phones was the significant reason for these accidents.[1] While in Pak-

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

[1] https://www.edgarsnyder.com/car-accident/cause-of-accident/cell-phone/cell-phone-+statistics.html

istan, around 15000 individuals passed on in road accidents.[2] Distracted driving was responsible for the death of 3477 individuals, while 391,000 were seriously injured. The most common cause of reported car accidents was texting or talking on mobile phones while driving [5]. While in Pakistan, about 15000 individuals passed on in road accidents.[3]

Distracted driving detection systems can be used to prompt early warnings to alarm drivers of hazardous driving conduct, including using a cell phone to call or text, using navigation applications, or selecting radio frequencies or music [6]. Distracted driving identification techniques are predominantly found in the driver's facial expression, head activity, line of sight, or body activity [7]. The driver's driving conduct and physiological state can be recognized through the visual following, target identification, movement acknowledgment, and different advancements. Detecting distracted driving has gained much attention from the research community, government agencies, and industry [8]–[10]. The driver's activity that diverts his attention during driving is a distraction like interaction with other passengers, making phone calls, and adjusting the multi-media and navigation tools. Most computer vision algorithms detect the driver's behavior using traditional computer vision and machine learning algorithms. We used deep learning algorithms to analyze the driver's abnormal behavior. Deep learning techniques have significantly improved the accuracy of vision-related tasks. Recent technological progress makes it possible for real-time algorithms to detect the distraction activity and assist and alert the distracted driver.

With improved deep learning classification and detection innovation, expanding analysts broke down driving conduct through Convolutional Neural Networks (CNNs). More scientists have additionally started to assemble their examination datasets. A combination of pre-prepared sparse filters and convolutional neural networks [11] was utilized to expand the arrangement precision of the Southeast University Driving Stance (SUE-DP) dataset to 99.78%. Authors in [12] improved the common areas with CNNs features (R-CNN) framework by replacing customary skin-like region extractor algorithms and scoring 97.76% on the SUE-DP dataset. Authors in [13] developed a dual-input deep three-dimensional convolutional network structure algorithm on a three-dimensional convolutional neural network (3DCNN), accomplishing 98.41% precision on the rail transit dataset. Authors in [14] prepared two independent convolutional neural networks by upgrading the size and number of convolution bits, which can adequately distinguish cell phones and hands, accomplishing 144 frames per second (fps) and 95.7% for cell phone precision utilization on the self-built dataset. In [15], a Multiscale Consideration CNN was proposed for recognizing driver actions.

Existing literature focuses on face location, hand/head recognition, eye movement investigating, and facial landmark detection. The definitions presented in guide[4] research in the field of distracted driving detection. It detects distractions that are manual, visual, or cognitive. This paper focuses on "manual" distractions using eight State Farm Distracted Driver Datasets (SDDD) classes, as manual distractions are primarily concerned with the driver's activities unrelated to safe driving. EfficientDet model is used to detect the objects involved in these distracting activities and the Region of Interest (ROI) of the body parts from the images. These results are better MAP compared to the previous detectors. In this paper, we use deep learning to detect and identify the distractions of a driver. A camera mounted above the dashboard captures RGB images. We use pre-trained networks on the ImageNet dataset in a "transfer learning" mode. We convert ten original SDDD into eight categories by combining calling (left/right) hands together, same as texting (left/right) hands. By cleaning the dataset, we put 1000 images in each and annotated every image from each class. EfficientDet model is used for detection purposes using the five variants, which accomplishes state-of-the-art accuracy while being up to 9x smaller utilizing 42x fewer FLOPs and less computation than earlier state-of-the-art detectors such as Yolo-V3, Faster R-CNN, RetinaNet, NAS-FPN [16]. EfficientNet surpasses state-of-the-art accuracy with up to 10x better efficiency, contrasts with previously commonly used backbones (i.e., ResNet, AmoebaNet). The proposed approach can help drivers maintain safe driving habits and reduce accidents.

The main contributions of this study are:

- Propose a model for distracted driver detection evaluated on the state farm distracted driver dataset.
- Implement five variants of the Efficientdet model to determine the most suitable model for driver distraction detection.
- Detect the objects and the region of interest of the body parts to detect distracted drivers.
- Results conclude that EfficientDet-D3 is the best model for detecting distracted drivers as it achieves a MAP of 99.16%.

The rest of the paper is arranged as follows. Section II reviews related work, Section III presents the methodology, Section IV presents experimental analysis and results. Finally, Section V concludes the paper.

II. RELATED WORK

Distracted conduct of the driver has been recognized utilizing two unique methodologies. One depends on the standard handcrafted features and then trains these disentangled features on the machine [17]. Such methods generally make use of the driver's eyes, head, and face to predict the unknown behavior of the driver. For example, authors in [18] distinguished the driver's distraction by estimating head rotations. First, they recognized the driver's head and then continuously

[2] https://nation.com.pk/10-Jul-2018/over-15000-people-die-in-pakistan-annually-due-to-traffic-accidents 3https://www.pbs.gov.pk/

[3] https://www.pbs.gov.pk/

[4] https://www.cdc.gov/motorvehiclesafety/distracted driving/

tracked its head movement. The Haar-wavelet Ada-boots cascades were used to recognize the head, and the localized gradient orientation was used to assess the posture. The main disadvantage of this calculation was that this only considers the driver's head pose, even though most activities to detect inactivity in the driver rely on facial features.

Authors in [13] present fine-tuned the CNN-based inception ResNet model concerning oneself gathered pictures of driver interruption action to improve accuracy. Their self-created dataset contains six classes, and every classification includes 1000 pictures. First, they apply the diverse preprocessing methods on train and test pictures. These preprocessing strategies incorporate cropping, adjustment, and flipping. After using the various preprocessing procedures, the information was passed to the Inception ResNet model for training reasons. The model is pre-trained on the ILSVRC 2012 dataset. They have accomplished 83% accuracy on the test information. Authors in [19] present a deep learning-based architecture that recognizes the inactiveness of the driver. They have utilized the highlights of the pretrained VGG-19 and fine-tuned them on the publicly arranged dataset. They had accomplished the best test precision of 95% and 80% exactness on the approval information of each class. They had likewise asserted that their model did not go towards the overfitting even on the oneself-produced dataset. They appeared in outcomes that their proposed model beats the XGBoost regarding accuracy by nearly 7%.

Authors in [20] present to apply SVM and fine-tune three CNN to the same combined features of those three networks to classify each edge into four classes: alert, nodding, drowsy with blinking, and yawning. However, the technique was tested on simulated data, where the signs of drowsiness were generally quickly visible, classifying on the apparent signs of drowsiness. That one was worth noting that the exactness reported in the analysis was 65.2%, which was lower than the precision of 73%. Authors in [21] proposed the CNN named as DarNet for inactive driver recognition. They created the dataset which identified the driver inactiveness. The dataset contains six distinct categories related to driver inactiveness. These categories are everyday driving, talking, texting, reading, Makeup, and eating. They fine-tuned the inception v3 module. The inception v3 module was first trained on the state from the Kaggle dataset for the inactive driver. They accomplished 87.02% precision on the proposed design. Authors in [22] proposed a CNN-based architecture by transforming the VGG-16 network, utilizing the Leaky ReLU activation function rather than ReLU, and applying different regularization strategies to adapt to the issue of overfitting. Thus, outcomes showed that the system accomplished 96.31% precision on the AUC Distracted Driver dataset.

Authors in [23] used the distinctive CNN models, including AlexNet, VGG-16, and ResNet152. For training purposes, they used the state farm distracted drivers' dataset. The dataset contains seven unique classes. Each class includes 3000 frames. The 2300 images from each class were used for training purposes. Moreover, 700 images were used for testing. They fine-tuned the above-depicted model on various variants of the ILSVRC dataset. They fine-tuned the AlexNet on the ILSVRC2012 dataset and accomplished 70% precision on the test data. Authors in [24] present a massive and publicly accessible certified drowsiness dataset. The method's principal was a Hierarchical Multiscale Long Short-Term Memory (HMLSTM) network, which prominent recognized blink features were dealt with in a grouping. An enormous and public genuine sleepiness dataset (RLDD), which contained 30 hours of video, detects signs of drowsiness. This benchmark strategy makes a precision of 80%, which was higher than human judgment. Authors in [25] present a deep learning-based model for detecting driver drowsiness in android applications. They created a model based on the disclosure of facial landmark points.

Authors in [26] proposed a driving-related recognition framework dependent on the deep CNN model. They utilized Kinect cameras to gather images of diverted drivers, and the raw images were prepared with a GMM-based division calculation. Then, at that point, CNN models were utilized as a double classifier which accomplishes 91% exactness. Authors in [27] proposed a deep learning-based technique for gaining diverted driving information furthermore, built an examination framework called DarNet, which accomplished a characterization exactness of 87.02% on their collected dataset. Authors in [28]–[30] proposed deep learning models for anomaly detection, intrusion detection, and botnet attack detection. Authors in [31] proposed Drive-net, a technique that utilizes a blend of a CNN and a random decision forest to classify driver images. Driver-net accomplished a recognition exactness of 95% on the Kaggle dataset. Authors in [32] proposed a driver interruption acknowledgment framework that utilized generative adversarial networks (GANs) and exhibited that generative models could produce images of drivers in various driving situations. Using these images to expand preparation improved the framework's image characterization execution by 11.45%.

Authors in [33] presented that both SM and UC cannot join and exit deftly. Because of odd occasions, SM might be inactively disengaged from UC. Nevertheless, this UC does not have the foggiest idea about the disconnect of SM, which keeps on observing the channel. Another conceivable case is that SM and UC have seen the harmful activities of RA; nobody can initiatively quit. The two conditions will bring about pointless energy expenses and potential risks. Reference [34] especially significant research point is the physiological signal encryption and secure transmission identified with the security assurance; some arising advancements give an essential reference.

Authors in [35] proposed a strategy for following the driver's facial region and utilized the yaw course point to gauge the driver's facial activity to recognize the driver's facial direction. Authors in [36] effectively fostered a driver's facial activity recognition framework dependent on binocular stereo vision, utilizing a hidden Markov model to foresee the driver's facial activity. Later, with the improvement of machine learning technology and general society of driving conduct datasets, expanding examines were added to dissect the driver's phone calling, drinking, eating, and other perilous driving practices. Southeast University Driving Stance (SUE-DP) dataset [37] was proposed in 2011. The examination gathered four categories of occupied driving practices: "getting a handle on the controlling wheel," "operating the shift lever," "eating," and "chatting on the phone." Using the SUE-DP dataset: Authors in [37] implemented the multiwavelet transform method and the multilayer perceptron (MLP) classifier to perceive four predefined driving stances and acquired a precision of 90.61%. Authors in [38] utilized support vector machines (SVM) classification for acquiring 94.25% precision.

Existing literature focused on face location, hand/head recognition, eye movement investigating, and facial landmark detection. It detects distractions that are manual, visual, or cognitive. Furthermore, the existing work of distracted driver detection is concerned with a limited set of distractions (mainly cell phone usage). At present, computer vision strategies are generally used to extract features, classify and detect images. Such arrangement tasks were completed by different deep neural network models running on high-performance computers to accomplish better acknowledgment precision.

III. PROPOSED FRAMEWORK

The proposed model contains two steps. In the first step, we perform preprocessing on the data. In the second step, we detect the objects involved in these distracting activities and the ROI of the body parts from the dataset's images.

A. PRE-PROCESSING OF DATASET

In April 2016, State Farm started the competition on the website named Kaggle.[5] The purpose of this competition is to generate images related to distracted driver behavior. We use the State Farm Distracted Driver Dataset (SDDD) to train our proposed algorithm. State Farm has collected the 2D images by placing the camera into the vehicle's dashboard. The purpose of these images is to generate the results on this data that will directly or indirectly help improve the stats of causalities due to distracted driver behavior. Table 1 presents the summary details of original SDDD.

The original SDDD has two folders containing 22400 training images and 79727 testing images. The image has a resolution of 640 × 480 pixels. The training folder includes a total of 10 categories, which are as follows; calling (left hand), calling (right hand), texting (left hand), texting (right hand), everyday driving, operating on radio, inactiveness, talking to a passenger, looking behind, and drinking. Each category contains different images, as shown in Table 2. Therefore, the data in the testing folder is unlabeled. We only use data in the training folder to evaluate our method.

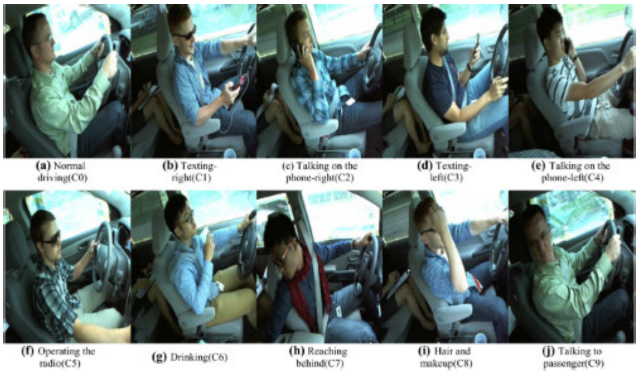
We convert ten original SDDD into eight categories by combining calling (left/right) hands together, same as texting (left/right) hands. By cleaning the dataset, we put 1000 images in each. We annotate every image from each class using the annotation tool labeling, and the annotated image is stored in a.xml file. Annotation is done to highlight the specific part of the images that include distracted objects and the region of interest of the body parts. There are 1000 annotated files in each class. So, this becomes 8000 RGB images and 8000 annotated files. The data distribution against each class has been shown in Tab. 2. We have split the dataset for each category into 80% train data and 20% validation data. The training set is split so that the validation set is not related to the training set. Some frames from the distracted driver dataset have been shown in Figure 1.

TABLE 1. Summary details of original SDDD.

SDDD classes	Description	Data (Frames)
c0	normal driving	2489
c1	texting (right hand)	2267
c2	calling (right hand)	2317
c3	texting (left hand)	2346
c4	calling (left hand)	2326
c5	operating on radio	2312
c6	drinking	2325
c7	looking behind	2592
c8	inactiveness	1911
c9	talking to passenger	2129

TABLE 2. Summary details of modified SDDD that utilized in this study.

Dataset classes	Images (Frames)	Annotated files
Normal driving	1000	1000
Texting (left/right hand)	1000	1000
Calling (left/right hand)	1000	1000
Operating on radio	1000	1000
Drinking	1000	1000
Looking behind	1000	1000
Inactiveness	1000	10001
Talking to passenger	1000	1000



(a) Normal driving(C0) (b) Texting-right(C1) (c) Talking on the phone-right(C2) (d) Texting-left(C3) (e) Talking on the phone-left(C4)

(f) Operating the radio(C5) (g) Drinking(C6) (h) Reaching behind(C7) (i) Hair and makeup(C8) (j) Talking to passenger(C9)

FIGURE 1. Samples of dataset to represent classes.

B. EFFICIENTDET

We use preprocessed and annotated data to train the *EfficientDet* model. After training the model, the video frame is input to predict whether the driver is distracted or not. We convert the image into the textual label related to our

[5] https://www.kaggle.com/c/state-farm-distracted-driver-detection/data

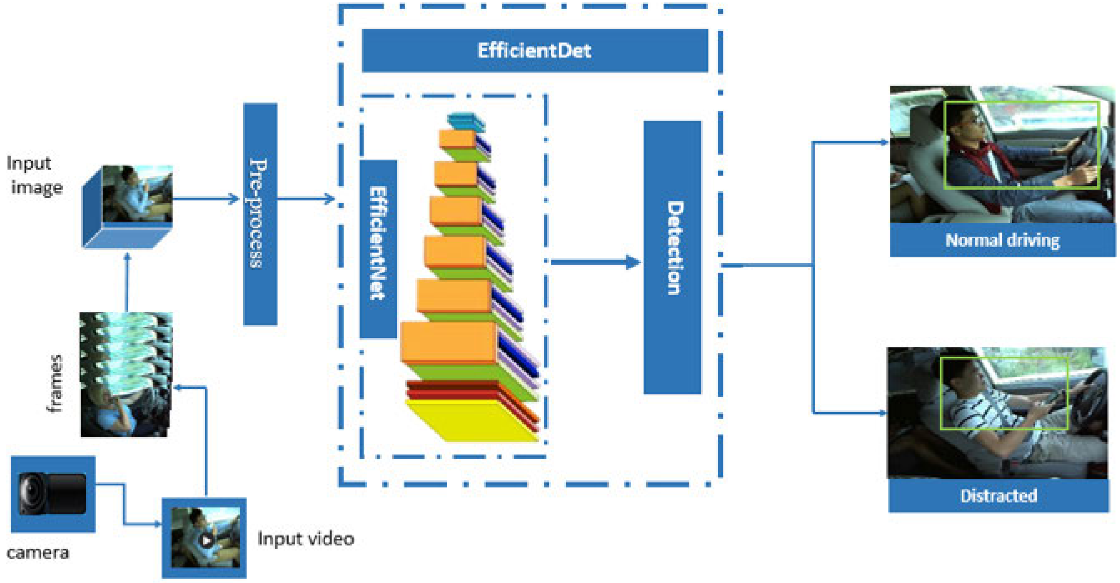


FIGURE 2. Proposed model overview.

defined image classification and detection categories. For image classification, we have used the transfer values of EfficientNet. Efficientdet model is used to detect the objects and the region of interest of body parts involved in these distracting activities to make accurate predictions and achieve state-of-the-art results. These steps are described as; extracting the convolution features of the input image, classifying the image, detecting the objects, and the ROI of the body parts, predicting by combining the label of classification and detection. The flow diagram of our methodology is shown in Figure 2.

Our proposed deep learning framework gets frames from the video streaming where the camera is placed on the vehicle's dashboard. These input images are passed through a preprocessing phase where each class image is annotated after image cleaning. Then fine-tune a pretrained ImageNet model EfficientNet for image classification. After that, objects are detected along with the region of interest of the body parts involved in these distracting activities using Efficientdet and give us the final result to detect distracted activities of the driver.

C. EFFICIENTDET ARCHITECTURE

The Google mind group developed the Efficientdet model. Improving the different dimensions includes combination construction of Feature Pyramid Networks (FPN) and acquiring thoughts from scaling technique of Efficientnet model, it is a model that can adapt recognition calculation. Efficientdet comprises three sections. The initial segment is prepared as ImageNet uses Efficientnet as the backbone network. The subsequent section is a Bidirectional Feature Pyramid Network (BiFPN), which performs the hierarchical and bottom-up, including various occasions for the yield normal for Levels $3-7$ in EfficientNet. The third component is the characterization and location box prediction network used to group and identify the diverted driver. Parts two and three modules can be rehashed repeatedly depending on hardware situations. In this paper, EfficientNet is used as the backbone model; with feature extraction of image contributions to the network and few feature map boundaries, rich data can be separated, partly guaranteeing location speed and precision. Input P3-P7 to BiFPN for include combination at that point. To acquire semantic data of various sizes, BiFPN receives weighted element combinations. Because Effificientdet is the objective recognition of anchor-based, the underlying anchor estimation should be changed appropriately to achieve better results. The architecture of Efficientnet is shown in Figure 3. BiFPN performs the function of a featured network. This takes features from the backbone network's levels $3-7$ and applies the BiFPN repeatedly. The combined features are fed into a class and box network to detect the object's class and bounding boxes.

1) BIDIRECTIONAL FEATURE PYRAMID NETWORK (BiFPN)

Conventional FPN accumulates multiscale attributes from through and through, as demonstrated in Figure 4(a) After convolution of the information highlight guide of layer 7, the yield highlight guide of layer seven is obtained. Convolving the combination highlight map obtained by up-testing the yield include a guide of layer seven and adding the information have a guide of layer six yields the yield highlight guide of layer six., convolving the combination highlight map obtained by up-examining the yield include a guide of layer four and adding the information highlight guide of layer three yields the yield highlight guide of layer 3. PANet improves FPN's performance element combination strategy,

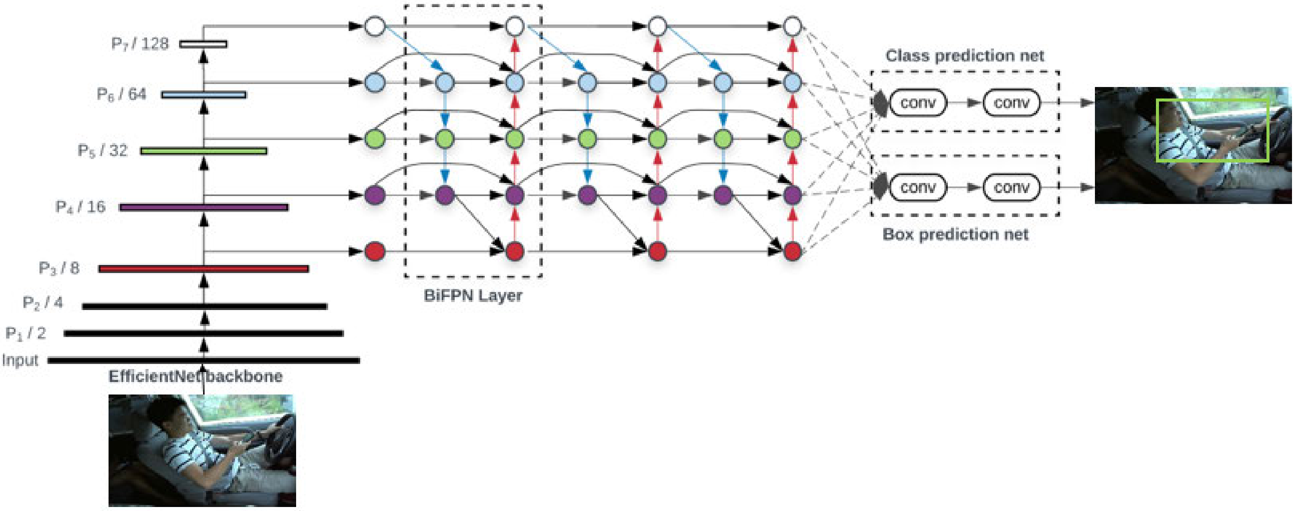


FIGURE 3. EfficientDet architecture.

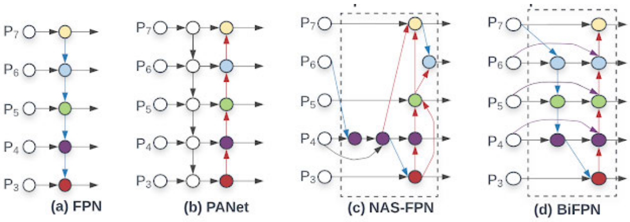


FIGURE 4. Feature network design.

as shown in Figure 4(b), receiving the hierarchical technique and the base up strategy. Figure 4(c) shows that an unpredictable trademark network geography, NAS-FPN, is discovered using the neural engineering search (NAS) technique. A significant amount of GPU processing time is required to achieve this result. The BiFPN showed in Figure 4(d) is improved in three ways:

- If a feature map only contains one piece of information, its commitment to feature combination is little, and it could be erased.
- At each level, associations are set up to allow users to join other features at a low cost.
- This is recommended that each BiFPN joins as a module and that the output of the previous BiFPN is used as the contribution of the following BiFPN. The circumstances determine the number of such structures required.

2) FEATURE FUSION METHODS

When the features of various scales are combined, the regular practice brings together the scales first and afterward adds the relating features. This expects that the heaviness of multiple features to the last combined component is similar. Indeed, various information features ought to contribute contrastingly to the last combination because of their diverse resolution. Unbounded fusion is calculated using Equation 1.

a: UNBOUNDED FUSION

$$O = \sum_i w_i.I_i \tag{1}$$

where learnable weight is wi that can be a per-channel, per-feature, or multi-dimensional tensor (per-pixel), the scalar weight is the most cost-effective way to calculate without sacrificing accuracy be used. The burden of such a weighting strategy was that there was no imperative weight to risk model preparation. SoftMax based fusion is calculated using Equation 2:

b: SOFTMAX BASED FUSION

$$O = \sum_i \frac{e_i^w}{\sum_j e_j^w}.I \tag{2}$$

This standardizes the heaviness of the past equation. However, this strategy does have the disadvantage of increasing the size of the computation. In addition, a quick combination approach is proposed to reduce the additional cost of inactivity. Fast normalized fusion is calculated using Equation 3.

c: FAST NORMALIZED FUSION

$$O = \sum_i \frac{w_i}{\epsilon + \sum_j w_j}.I \tag{3}$$

$w_i >= 0$ is guaranteed by implementing a Relu every after w_i, but rather $\epsilon = 0001.0$ has been a little worth to avoid mathematical flimsiness. The removal study reveals that the aftereffects of this weighting technique are similar to that of softmax include combination, with the computing speed on GPU improved by 30%.

3) COMPOUND SCALING

Another compound scaling strategy for object detection was proposed due to the compound scaling used in EfficientNets. This method employs a coefficient ϕ to simultaneously scale up all backbone network components, BiFPN network, class/box network, and resolution. The scaling of each network segment is shown below:

a: BACKBONE NETWORK

Use the same coefficients as defined in efficientdet-B0 to efficientdet-B6 to reuse their ImageNet pre-prepared loads.

b: BiFPN

The width (channels) is dramatically developed, and the depth is straightly expanded (layers). The width and depth are officially scaled using the Equation 4.

$$W_{bifpn} = 6(1.3^{\phi}),\ bifpn = 3 + \phi \tag{4}$$

c: BOX/CLASS PREDICTION NETWORK

The width is fixed to be the same as in the BiFPN, but the depth is straight and can be calculated using Equation 5.

$$D_{box} = D_{class} = 3 + (\frac{\phi}{3}) \tag{5}$$

d: INPUT IMAGE RESOLUTION

The resolution is expanded directly because it should be dividable by $2^7 = 128$. This is accomplished through the following Equation 6:

$$R_{input} = 512 + \phi.128 \tag{6}$$

IV. EXPERIMENTAL ANALYSIS AND RESULTS

To train the Effificientdet model, we first annotate data. The annotation of the data means we highlight the specific object on the dataset's images. We divided the data into two parts for training purposes: 80% training data and 20% test data. The Effificientdet model is trained on Nvidia GPU using PyCharm frame 2020 of version 1.4.0 and Python 3.6. The experimental system is Linux Ubuntu 16.04, and the graphics card is GeForce GTX 1080Ti 11GB. The software used the Windows 10 operating system, Keras, and TensorFlow deep learning framework. Next, we fine-tune the pre-trained model Efficientdet at epoch 50 with steps size 250, batch size 4, threshold 0.3, and learning rate (1e-3).

A. LOSS AND MAP

The experimental results of different variants of the Efficientdet pretrained model are as follows. The losses and mAp of Efficientdet-D0, Efficientdet-D1, Efficientdet-D2, Efficientdet-D3, and Efficientdet-D4 are presented in Figure 5 and Figure 6, respectively. Among the five variants of the pretrained model Efficientdet, Efficientdet-D3 achieves the highest MAP and the lowest loss. Efficientdet-D0 has the lowest MAP and also the most considerable loss. For the task of distracting behavior detection, Efficientdet-D3 achieves a MAP of 99.16%. Efficientdet-D4 has the second-largest MAP, which is 98.89%. Thus, the above-pretrained model can detect distracted driving behaviors in the State Farm dataset. However, the pretrained model Efficientdet also has a small value of losses; hence, Efficientdet can detect images of distracted drivers without this specific dataset. Each network contains eight models and the final layers. After this, each of them contains seven blocks. These blocks further have varying sub-blocks whose number is increased as we move from EfficientNetB0 to EfficientNetB7. The total number of layers in EfficientNet-B0 is 237, and in EfficientNet-B7, the total comes out to 813. EfficientNet-D3 has more number parameters, and several feature maps (channels) vary, increasing the number of parameters. This tuned version is the reason it is performing well.

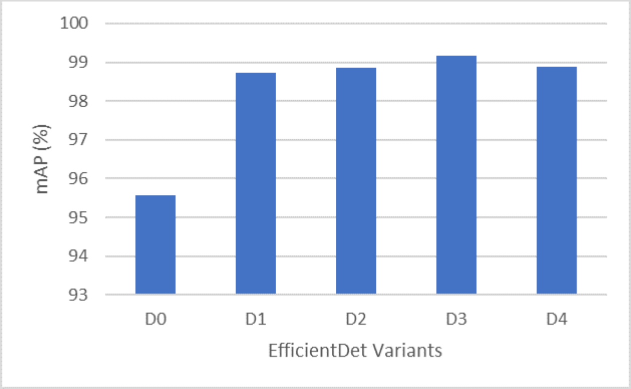


FIGURE 5. **Efficient variants MAP (%).**

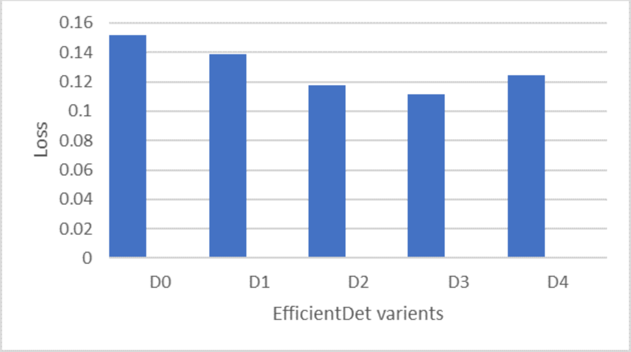


FIGURE 6. **Loss function of Efficientdet variants.**

The model checkpoints are automatically saved in a temporary directory during training. Running the training script can restart training from the most recent checkpoint in this temporary directory. Training will resume from the original pre-trained checkpoint that we saved in the efficientdet folder if none are found. As a result, if we intend to retrain from the original pre-trained model, make sure you delete this temporary folder. To avoid strange errors significantly, if we change some hyper-parameters and restart the fine-tuning process, we should delete this temporary directory.

TABLE 3. MAP and loss of EfficientDet-D0 to EfficientDet-D4.

Model	MAP (%)	loss
Efficientdet-D0	95.56	0.1515
Efficientdet-D1	98.74	0.1389
Efficientdet-D2	98.86	0.1176
Efficientdet-D3	99.16	0.1114
Efficientdet-D4	98.89	0.1243

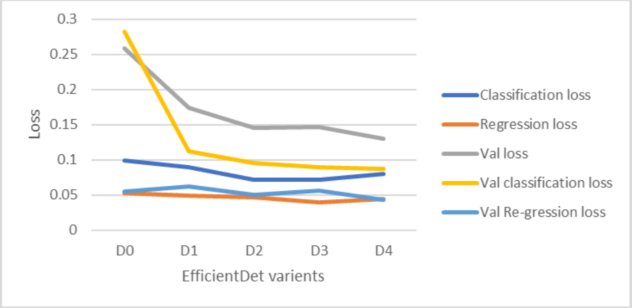


FIGURE 7. Losses of EfficientDet-D0 to EfficientDet-D4.

Table 3 shows the mean average precision and loss rate of EfficientDet-D0 to EfficientDet-D4. MAP compares the ground-truth bounding box to the detected box and returns a score. However, the model is more accurate in its detection if the score is higher. From Efficientdet-D0 to Efficientdet-D3 MAP progressively increases with epoch 50, EfficientDet-D3 has 99.16 MAP. Whereas at Efficientdet-D4 MAP decrease to 98.89 from Efficientdet-D3 same as from Efficientdet- D0 to Efficientdet-D3 loss gradually decreases. Efficientdet-D3 has a loss rate of 0.1114 whereas, at Efficientdet-D4, loss increases to 0.1243 from Efficientdet-D3. Hence, it shows that Efficientdet-D3 has a high MAP and low loss rate, so it is the best detection method.

Table 4 shows the evaluation results of EfficientDet-D0 to EfficientDet-D4. Efficientdet-D2 shows the lowest classification loss, 0.0713, then others whereas, the classification loss of D3 is 0.0001 higher than D2. Efficientdet-D3 has the lowest regression loss, 0.04, then other variants. The validation loss of Efficientdet-D4 is 0.1302, the lowest validation loss of different variants. Efficientdet-D3 has a lesser validation classification loss of 0.0901 than other Efficientdet model variants shown in Tab. 4. EfficientdetD4 has the lowest validation regression loss, 0.0434, compared to Efficientdet (D0, D1, D2, and D3). These results are shown in graphical form in Figure 7.

B. AVERAGE PRECISION

Table 5 shows the average precision of all eight classes of the dataset used in Efficientdet five variants which are Efficientdet-D0, Efficientdet-D1, Efficientdet-D2, Efficientdet-D3, and Efficientdet-D4 of model Efficientdet at the last epoch 50. Efficientdet-D3 achieves the highest average precision at calling class 0.9936. Efficientdet-D0 achieves the lowest average precision at calling class 0.9761. Efficientdet-D4 achieves the highest average precision at drinking class 0.9964. Efficientdet-D0 achieves the lowest average precision at drinking class 0.9805. Efficientdet-D2 achieves the highest average precision at inactiveness class 0.9718. Efficientdet-D0 achieves the lowest average precision at inactiveness class 0.8279. Efficientdet-D4 achieves the highest average precision at looking-behind class 0.9999. Efficientdet-D1 achieves the lowest average precision at looking-behind class 0.9945. Efficientdet-D4 achieves the highest average precision at normal-driving class 0.9942. Efficientdet-D0 achieves the lowest average precision at normal-driving class 0.9513. Efficientdet-D3 achieves the highest average precision at operating-on-radio class 0.9995. Efficientdet-D0 achieves the lowest average precision at operating-on-radio class 0.9912. Efficientdet achieves the highest average precision at talking-to-passenger class 0.9996. Efficientdet-D2 achieves the lowest average precision at talking-to-passenger class 0.9055. Efficientdet-D2 achieves the highest average precision at texting class 0.9989. Efficientdet-D0 achieves the lowest average precision at texting class 0.9922.

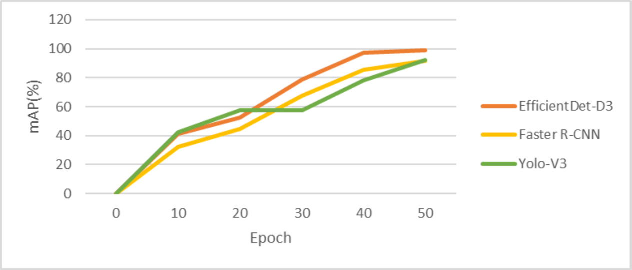


FIGURE 8. Comparison of state-of-art detectors.

C. COMPARISON OF DETECTION MODELS

We train our dataset on three detection models: Efficientdet, Faster R-CNN, and Yolo-V3. As we notice that Efficientdet has the highest MAP, so we compare Efficientdet-D3 with Faster-RCNN and Yolo-V3.

Figure 8 shows that Efficientdet-D3 has the highest MAP, 99.16 at final epoch 50, and the lowest MAP, 41.46 at epoch 10. On the other hand, faster R-CNN has the highest MAP 91.45 at the last epoch 50 and the lowest MAP 32.46 at epoch 10. On the other hand, Yolo-V3 has the highest MAP 92.15 at final epoch 50 and the lowest MAP 42.34 at epoch 10. This shows that Efficientdet-D3 has the highest MAP 99.16 than Faster R-CNN and Yolo-V3; hence Efficientdet-D3 is the best model for detection.

D. TEST RESULTS

The retrained model is saved as a checkpoint file containing only the model weights, not the model architecture. To perform inference with our fine-tuned model, we must first export the model into the saved model (.pb format). It is worth noting that the pre-processing and post-processing operations will be incorporated into the exported model. As a result,

TABLE 4. Results of EfficientDet-D0 to EfficientDet-D4.

Model	Classification loss	Regression loss	Val loss	Val classification loss	Val Regression loss
Efficientdet-D0	0.0994	0.0521	0.2584	0.2829	0.0555
Efficientdet-D1	0.0893	0.0496	0.1745	0.1128	0.0625
Efficientdet-D2	0.0713	0.0463	0.1455	0.0958	0.0497
Efficientdet-D3	0.0714	0.04	0.1463	0.0901	0.0562
Efficientdet-D4	0.0805	0.0438	0.1302	0.0868	0.0434

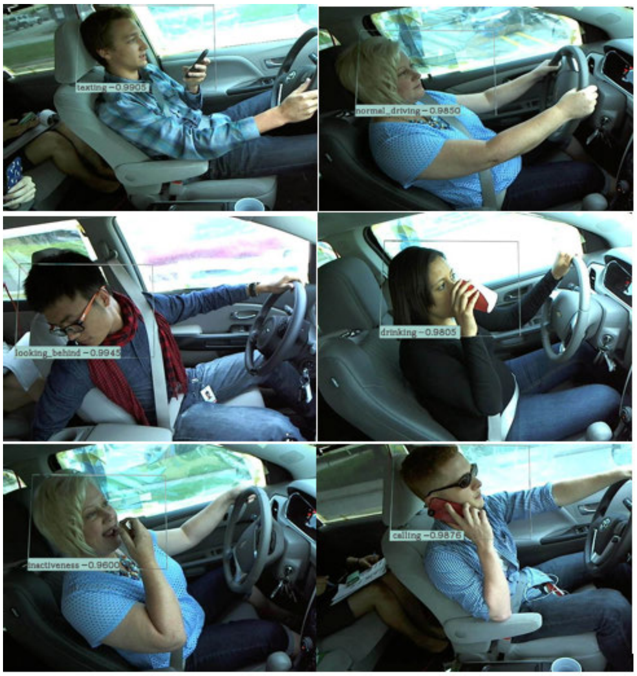


FIGURE 9. Test results of EfficientDet-D0 to EfficientDet-D4.

TABLE 5. Average precision of all classes.

	AP				
Classes	D0	D1	D2	D3	D4
Calling	0.9761	0.9850	0.9817	0.9936	0.9876
Drinking	0.9805	0.9895	0.9869	0.9913	0.9964
Inactiveness	0.8279	0.96	0.9718	0.9611	0.9492
Looking behind	0.9963	0.9945	0.9996	0.9981	0.9999
Normal driving	0.9513	0.9829	0.9850	0.9903	0.9942
Operating on radio	1	0.9991	1	1	1
Talking to passanger	0.9286	0.9921	0.9055	0.9996	0.9857
Texting	0.9922	0.9961	0.9989	0.9985	0.9980

some parameters should be specified during the exporting process, such as the minimum score threshold to filter out low confidence-bound boxes. Finally, we apply the model to the distracted driver dataset we prepared and decode the prediction image shown in Figure 9.

Efficientdet-D0 to Efficientdet-D3 MAP progressively increases with epoch 50, EfficientDet-D3 has a high MAP 99.16, and less loss 0.1114 than other running speeds is 30% on faster GPU. The final EfficientDet-D4 has low MAP and high loss than Efficientdet-D3, which shows that EfficientDet-D3 is the best model for detection purposes.

V. CONCLUSION

In this paper, the Efficientdet detection model is used to efficiently detect distracted drivers and the ROI of the body parts to reduce serious incidents. Our analysis suggests that EfficientNet compound scaling must simultaneously increase depth, width, and resolution when designing more extensive networks. As a result, it has a faster running speed and a higher success rate than previous object detection models. The development of EfficientDet is highly beneficial to object detection research and the development of a wide range of applications. Experimental results show that the proposed approach outperforms existing methods and conclude that EfficientDet-D3 is the best model for detecting distracted drivers as it achieves Mean Average Precision (MAP) of 99.16% along with learning rate (le-3), epoch 50, batch size 4, and step size 250, demonstrating that it can potentially help drivers maintain safe driving habits. It is expected to impact the advancement of computer vision significantly.

In the future other two variants, Efficientdet-D5 and Efficientdet-D6 of model Efficientdet can be used for detection purposes. Furthermore, an extended version of the dataset can be presented to improve the distraction detection system by incorporating additional sensing modalities. For example, we can use a microphone to record the sound and voice in the car, which provides valuable clues for detecting various distracted driving behaviors. In addition, we can improve the models by using dynamic data.

REFERENCES

[1] C. Cher, M. Dan, H. Karisa, W. Madonna, and M. Raby, "Using naturalistic driving data to assess the prevalence of environmental factors and driver behaviors in teen driver crashes," Tech. Rep., 2015.

[2] *Global Health Estimates 2015: Deaths by Cause, Age, Sex, by Country and by Region 2000–2015*, World Health Organization, Geneva, Switzerland, 2016.

[3] *World Health Statistics 2017: Monitoring Health for the SDGS, Sustainable Development Goals*, World Health Organization, Geneva, Switzerland, 2017.

[4] Y. Abouelnaga, H. M. Eraqi, and M. N. Moustafa, "Real-time distracted driver posture classification," 2017, *arXiv:1706.09498*.

[5] *2015 Motor Vehicle Crashes: Overview*, Traffic Safety Facts, Research Note, National Highway Traffic Safety Administration, Washington, DC, USA, 2016, pp. 1–9.

[6] D. Liu, "Driver status monitoring and early warning system based on multi-sensor fusion," in *Proc. Int. Conf. Intell. Transp., Big Data Smart City (ICITBS)*, Jan. 2020, pp. 24–27.

[7] Y. Liu, Y. Zhang, J. Li, J. Sun, F. Fu, and J. Gui, "Towards early status warning for driver's fatigue based on cognitive behavior models," in *Proc. Int. Conf. Digit. Hum. Modeling Appl. Health, Saf., Ergonom. Risk Manage.*, Springer, 2013, pp. 55–60.

[8] X. Chen, C. Wu, Z. Liu, N. Zhang, and Y. Ji, "Computation offloading in beyond 5G networks: A distributed learning framework and applications," *IEEE Wireless Commun.*, vol. 28, no. 2, pp. 56–62, Apr. 2021.

[9] K.-L.-A. Yau, H. J. Lee, Y.-W. Chong, M. H. Ling, A. R. Syed, C. Wu, and H. G. Goh, ''Augmented intelligence: Surveys of literature and expert opinion to understand relations between human intelligence and artificial intelligence,'' *IEEE Access*, vol. 9, pp. 136744–136761, 2021.

[10] F. Rasheed, K.-L.-A. Yau, R. M. Noor, C. Wu, and Y.-C. Low, ''Deep reinforcement learning for traffic signal control: A review,'' *IEEE Access*, vol. 8, pp. 208016–208044, 2020.

[11] C. Yan, F. Coenen, and B. Zhang, ''Driving posture recognition by convolutional neural networks,'' *IET Comput. Vis.*, vol. 10, no. 2, pp. 103–114, 2016.

[12] S. Yan, Y. Teng, J. S. Smith, and B. Zhang, ''Driver behavior recognition based on deep convolutional neural networks,'' in *Proc. 12th Int. Conf. Natural Comput., Fuzzy Syst. Knowl. Discovery (ICNC-FSKD)*, Aug. 2016, pp. 636–641.

[13] W. Kim, H.-K. Choi, B.-T. Jang, and J. Lim, ''Driver distraction detection using single convolutional neural network,'' in *Proc. Int. Conf. Inf. Commun. Technol. Converg. (ICTC)*, Oct. 2017, pp. 1203–1205.

[14] C. Jin, Z. Zhu, Y. Bai, G. Jiang, and A. He, ''A deep-learning-based scheme for detecting driver cell-phone use,'' *IEEE Access*, vol. 8, pp. 18580–18589, 2020.

[15] Y. Hu, M. Lu, and X. Lu, ''Feature refinement for image-based driver action recognition via multi-scale attention convolutional neural network,'' *Signal Process., Image Commun.*, vol. 81, Feb. 2020, Art. no. 115697.

[16] M. Tan, R. Pang, and Q. V. Le, ''EfficientDet: Scalable and efficient object detection,'' in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 10781–10790.

[17] K. Kim, W. Lee, Y. Kim, H. Hong, E. Lee, and K. Park, ''Segmentation method of eye region based on fuzzy logic system for classifying open and closed eyes,'' Tech. Rep., 2015.

[18] E. Murphy-Chutorian and M. M. Trivedi, ''Head pose estimation and augmented reality tracking: An integrated system and evaluation for monitoring driver awareness,'' *IEEE Trans. Intell. Transp. Syst.*, vol. 11, no. 2, pp. 300–311, Jun. 2010.

[19] K. Arief, M. B. Safaa, O. Chaojie, and K. Fakhri, *ETE: End-to-End Deep Learning for Driver Distraction Recognition*. Springer, 2017.

[20] S. Park, F. Pan, S. Kang, and C. D. Yoo, ''Driver drowsiness detection system based on feature representation learning using various deep networks,'' in *Proc. Asian Conf. Comput. Vis.*, Springer, 2016, pp. 154–164.

[21] C. Streiffer, R. Raghavendra, T. Benson, and M. Srivatsa, ''A deep learning solution for distracted driving detection,'' in *Proc. 18th ACM/IFIP/USENIX Middleware Conf., Ind. Track*, 2017, pp. 22–28.

[22] B. Baheti, S. Gajre, and S. Talbar, ''Detection of distracted driver using convolutional neural network,'' in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2018, pp. 1032–1038.

[23] M. Hssayeni, S. Saxena, R. Ptucha, and A. Savakis, ''Distracted driver detection: Deep learning vs handcrafted features,'' *Electron. Imag.*, vol. 2017, no. 10, pp. 20–26, Jan. 2017.

[24] R. Ghoddoosian, M. Galib, and V. Athitsos, ''A realistic dataset and baseline temporal model for early drowsiness detection,'' in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2019, pp. 178–187.

[25] R. Jabbar, K. Al-Khalifa, M. Kharbeche, W. Alhajyaseen, M. Jafari, and S. Jiang, ''Real-time driver drowsiness detection for Android application using deep neural networks techniques,'' *Proc. Comput. Sci.*, vol. 130, pp. 400–407, Jan. 2018.

[26] W. Zhang, B. Cheng, and Y. Lin, ''Driver drowsiness recognition based on computer vision technology,'' *Tsinghua Sci. Technol.*, vol. 17, no. 3, pp. 354–362, Jun. 2012.

[27] M. Shahverdy, M. Fathy, R. Berangi, and M. Sabokrou, ''Driver behavior detection and classification using deep convolutional neural networks,'' *Expert Syst. Appl.*, vol. 149, Jul. 2020, Art. no. 113240.

[28] A. R. Javed, S. U. Rehman, M. U. Khan, M. Alazab, and T. Reddy, ''CANintelliIDS: Detecting in-vehicle intrusion attacks on a controller area network using CNN and attention-based GRU,'' *IEEE Trans. Netw. Sci. Eng.*, vol. 8, no. 2, pp. 1456–1466, Apr. 2021.

[29] A. R. Javed, M. Usman, S. U. Rehman, M. U. Khan, and M. S. Haghighi, ''Anomaly detection in automated vehicles using multistage attention-based convolutional neural network,'' *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 7, pp. 4291–4300, Jul. 2021.

[30] A. R. Javed, Z. Jalil, S. A. Moqurrab, S. Abbas, and X. Liu, ''Ensemble AdaBoost classifier for accurate and fast detection of BotNet attacks in connected vehicles,'' *Trans. Emerg. Telecommun. Technol.*, p. e4088, 2020.

[31] M. S. Majdi, S. Ram, J. T. Gill, and J. J. Rodriguez, ''Drive-Net: Convolutional network for driver distraction detection,'' in *Proc. IEEE Southwest Symp. Image Anal. Interpretation (SSIAI)*, Apr. 2018, pp. 1–4.

[32] C. Ou and F. Karray, ''Enhancing driver distraction recognition using generative adversarial networks,'' *IEEE Trans. Intell. Vehicles*, vol. 5, no. 3, pp. 385–396, Sep. 2020.

[33] J. Song, Q. Zhong, W. Wang, C. Su, Z. Tan, and Y. Liu, ''FPDP: Flexible privacy-preserving data publishing scheme for smart agriculture,'' *IEEE Sensors J.*, vol. 21, no. 16, pp. 17430–17438, Aug. 2021.

[34] L. Zhang, Z. Zhang, W. Wang, Z. Jin, Y. Su, and H. Chen, ''Research on a covert communication model realized by using smart contracts in blockchain environment,'' *IEEE Syst. J.*, early access, Mar. 9, 2021, doi: 10.1109/JSYST.2021.3057333.

[35] X. Liu, Y. Zhu, and K. Fujimura, ''Real-time pose classification for driver monitoring,'' in *Proc. IEEE 5th Int. Conf. Intell. Transp. Syst.*, Sep. 2002, pp. 174–178.

[36] H. Eren, U. Celik, and M. Poyraz, ''Stereo vision and statistical based behaviour prediction of driver,'' in *Proc. IEEE Intell. Vehicles Symp.*, Jun. 2007, pp. 657–662.

[37] C. H. Zhao, B. L. Zhang, J. He, and J. Lian, ''Recognition of driving postures by contourlet transform and random forests,'' *IET Intell. Transp. Syst.*, vol. 6, no. 2, pp. 161–168, 2012.

[38] C. Zhao, B. Zhang, J. Lian, J. He, T. Lin, and X. Zhang, ''Classification of driving postures by support vector machines,'' in *Proc. 6th Int. Conf. Image Graph.*, Aug. 2011, pp. 926–930.

FAIQA SAJID is currently pursuing the M.S. degree with the Department of Computer Science, Kinnaird College for Woman, Lahore, Pakistan. Her research interests include machine learning, wireless sensor networks, mobile computing, and security issues in mobile cloud computing.

ABDUL REHMAN JAVED (Member, IEEE) received the master's degree in computer science from the National University of Computer and Emerging Sciences, Islamabad, Pakistan. He worked with the National Cybercrimes and Forensics Laboratory, Air University, Islamabad, where he is currently a Lecturer with the Department of Cyber Security. He is also a Cybersecurity Researcher and a Practitioner with industry and academic experience. He has reviewed over 150 scientific research articles for various well-known journals. He has authored over 50 peer-reviewed research articles and is supervising/co-supervising several graduate (B.S. and M.S.) students on topics related to health informatics, cybersecurity, mobile computing, and digital forensics. His current research interests include, but are not limited to, mobile and ubiquitous computing, data analysis, knowledge discovery, data mining, natural language processing, smart homes and their applications in human activity analysis, human motion analysis, and e-health. He aims to contribute to interdisciplinary research of computer science and human-related disciplines. He is a member of ACM. He is also a TPC Member of the Fourth International Workshop on Cybercrime Investigation and Digital forensics-CID 2021 and the 44th International Conference on Telecommunications and Signal Processing. He has served as a Moderator for the First IEEE International Conference on Cyber Warfare and Security (ICCWS).

ASMA BASHARAT joined the Department of Computer Science, Kinnaird College for Women, Lahore, as an Assistant Professor. She has authored or coauthored several peer-reviewed papers in professional journals and the proceedings of conferences. Her research interests include the areas of machine learning algorithms, wireless sensor networks, mobile computing, self-organized networks, big data analytics, and the Internet of Things.

NATALIA KRYVINSKA received the Ph.D. degree in electrical and IT engineering from the Vienna University of Technology, Austria, and the Docent (Habilitation) degree in management information systems from Comenius University in Bratislava, Slovakia. She is currently a Full Professor and the Head of the Information Systems Department, Faculty of Management, Comenius University in Bratislava. Previously, she served as a University Lecturer and a Senior Researcher with the eBusiness Department, School of Business Economics and Statistics, University of Vienna. She received the Professor Title and was appointed for the professorship by the President of the Slovak Republic. Her research interests include complex service systems engineering, service analytics, and applied mathematics.

ADIL AFZAL 's research interests include, but are not limited to, computer forensics, machine learning, criminal profiling, software watermarking, intelligent systems, and data privacy protection.

MUHAMMAD RIZWAN received the M.Sc. degree from PUCIT, Lahore, Pakistan, in 2006, the M.S. degree from CIIT, Lahore, in 2012, and the Ph.D. degree from HUST, Wuhan, China, in 2017. In 2017, he joined the Department of Computer Science, Kinnaird College for Women, Lahore, as an Assistant Professor. He has authored or coauthored several peer-reviewed papers in professional journals and the proceedings of conferences. His research interests include the areas of machine learning algorithms, wireless sensor networks, mobile computing, self-organized networks, big data analytics, and the Internet of Things.