IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received November 4, 2021, accepted December 5, 2021, date of publication December 7, 2021,
date of current version December 29, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3133797

Robust Deep Learning-Based Driver Distraction Detection and Classification

AMAL EZZOUHRI[1,2], ZAKARIA CHAROUH[1,2], MOUNIR GHOGHO[1,3], (Fellow, IEEE), AND ZOUHAIR GUENNOUN[2], (Senior Member, IEEE)

[1]TIC Laboratory, International University of Rabat, Rabat 11103, Morocco
[2]ERSC Team, Mohammadia School of Engineering, Mohammed V University, Rabat 10056, Morocco
[3]Faculty of Engineering, School of Electronic and Electrical Engineering, University of Leeds, Leeds LS2 9JT, U.K.

Corresponding author: Amal Ezzouhri (amal.ezzouhri@uir.ac.ma)

This work was supported in part by the National Agency for Road Safety (NARSA) of the Moroccan Ministry of Equipment, Transport, Logistics and Water, through the National Center for Scientific and Technical Research (CNRST).

ABSTRACT Driver distraction is a major cause of road accidents. Distracting activities while driving include text messaging and talking on the phone. In this paper, we propose a robust driver distraction detection system that extracts the driver's state from the recordings of an onboard camera using Deep Learning. We consider ten driving activities, which consist of one normal driving and nine distracted driving behaviors. Nine drivers were included in the experiments, and each one was asked to perform the ten activities in naturalistic and simulated driving situations. The main feature of the proposed solution is the extraction of the driver's body parts, using deep learning-based segmentation, before performing the distraction detection and classification task. Experimental results show that the segmentation module significantly improves the classification performance. The average accuracy of the proposed solution exceeds 96% on our dataset and 95% on the public AUC dataset.

INDEX TERMS Driver distraction, advanced driver assistance systems, deep convolutional neural network, driving behavior, semantic segmentation.

I. INTRODUCTION

According to the National Highway Traffic Safety Administration (NHTSA) [1], 80% of accidents and 16% of highway deaths are the result of distracted driving. The National Safety Council (NSC) [2] estimates that 1.6 million (25%) (resp. 1 million (18%)) of annual crashes are due to the use of the phone (resp. to text messaging) during driving. Other sources of distraction while driving include eating, drinking, adjusting the radio, reaching for objects in the car.

Although the issue of driver distraction is not new, it has worsened significantly with the advent of smartphones. Distracted driving can be divided into three types: visual (taking eyes off the road, e.g., looking for items on the floor of the car, reaching behind), manual (taking hands off the steering wheel, e.g., eating, drinking), and cognitive (driver losing focus while driving, e.g., talking, mind away from driving). The three types of distraction can lead to more considerable lane variations, a lower capability of vehicle control, a slower response to hazards, and a less efficient perception of the road environment than attentive driving. It is worth pointing out that some activities fall under a combination of two or three distracted driving categories. These activities are hazardous, as they tend to take more attention away from the driving task than activities that fall under one category only. Texting is one of such activities. Indeed, while texting, the drivers are distracted visually, as they look at their phones instead of the road and nearby cars, manually, as they type messages instead of keeping their hands on the steering wheel and being ready to react, and cognitively, as they concentrate on the conversations instead of the situations unfolding in their driving environment.

Many efforts have been made in the literature to detect driver distraction, especially machine vision-based techniques. Most of them use the whole image as input to the CNN-based classifiers. However, as humans, we focus only on relevant regions from the driver when performing this task. We assumed that we could help classification models to learn and predict the exact distraction type if we selected only those critical regions from the image and used them as

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

input. Thus, we propose using a robust deep learning-based human body part segmentation method as a preprocessing step before the CNN-based classification model. Our work aims to design an effective framework to correctly recognize distraction activities of any driver from the video captured by an on-board camera. The main contributions of this paper can be summarized as follow:

- We propose a new annotated *naturalistic driving* image dataset for the study of driver distraction detection of more than 38K images, which we will share with the scientific community.
- We propose an end-to-end deep learning-based driver distraction detection system able to be used with any driver and in any environment.
- We propose using deep learning-based human body parts segmentation method to efficiently remove irrelevant objects and identify the driver's critical body parts (i.e., the image regions that contribute to the driver's distraction recognition). We show that this significantly improves classification accuracy. To the best of our knowledge, a deep learning-based body segmentation has not been used before for driver distraction detection.

The rest of the paper is organized as follows. In Section II, we discuss the most important datasets used to detect driver distractions, we review the methods that have been proposed in the literature for the detection of driver distractions, and we discuss the relevant human segmentation methods. In Section III, we describe the data acquisition system used to build our dataset. The proposed framework for the detection and classification of driving distractions is described in Section IV. The experimental results and their analysis are presented in Section V. Finally, conclusions are drawn in Section VI.

II. RELATED WORK

This section reviews some of the relevant existing work on driver detection. In the literature, the issue of driver detection has been addressed using either natural driving conditions or a driving simulator. Distraction detection systems differ in the type of data used, and the approach used to build the detection model. Three main types of data sources have been used, as described next.

A. DATA SOURCES

- *Visual data*: these include facial expressions, eye gaze, and body movements, captured using an onboard camera. In [3]–[5], the authors used pre-trained networks to extract driving features and recognize behaviors from images. In [6], the authors provided a system that detects driver fatigue using images of the driver's face; they extracted facial components, including eyes, mouth, and head position from the input image. The extracted facial components are used to train the classifier model (based on the Support Vector Machine method) to predict whether the driver is in danger or not. In [7], the authors proposed an algorithm for the detection of manual distractions. The detection algorithm consists of two modules; the former predicts the bounding boxes of the driver's right hand and right ear from the RGB images. While the second takes the bounding boxes as input and predicts the type of distraction. Five types of non-driving tasks were taken into account: talking on the cell phone, texting, drinking water, using the touchscreen, and placing a marker in the cup-holder. The algorithm was evaluated on videos collected using a driving simulator and obtained an F1 score of 0.74. In the experimental setup of this paper, the five non-driving activities mentioned above may not only be associated with movements of the right hand and right ear; the drivers' left hands can be engaged in one of the five activities while keeping the right hand in the steering wheel. Consequently, the proposed algorithm falsely considers it as safe driving.
- *Vehicle control data*: these include different vehicle system data such as acceleration, deceleration, steering angle, vehicle heading angle, speed, steering input count, etc. In [8], the authors developed an algorithm that can detect texting and eating/drinking tasks using vehicle control data collected by a driving simulator. In [9], the authors provided a system that recognizes three cognitive distractions states - no cognitive distraction, low cognitive distraction, and high cognitive distraction - using different vehicle control data.
- *Physiological data*: these include electrocardiogram (ECG) and electroencephalogram (EEG) data. As mentioned previously, three general types of distractions may occur while driving (i.e., visual, manual, and cognitive). Visual and manual distractions can be easily recognized based on visual sensors (i.e., cameras), unlike cognitive distractions, which are not easily detected and require a robust data fusion system. The challenge is to integrate different data streams, including visual, vehicle control, and physiological data. In [10], the authors developed an algorithm to identify cognitive distraction based on eye gaze, head movements, changes in pupil diameter, and ECG heart rate.

Researchers often build datasets to carry out their studies, but rarely publish these datasets. The dataset of StateFarm on Kaggle [11] was the first publicly available dataset, but used for competition purposes only. The dataset consists of examples of ten distractions: safe driving, texting using right hand, talking on the phone using right hand, texting using left hand, talking on the phone using left hand, operating the radio, drinking, reaching behind, doing hair and makeup, and talking to a passenger. In [12], the authors built a new Distracted Driver dataset which is similar to the StateFarm's dataset (i.e. it is composed of the same ten distraction activities). A total of 31 volunteers from seven different countries participated in the creation of this dataset. However, we have found that this dataset is not balanced (e.g. the normal driving class represents 21% of the complete data against only 7% for the reaching behind class). Moreover, some drivers did not participate in *all* distraction activities.

TABLE 1. Estimation of incorrect annotation rates based on 2000 randomly selected (200 image per class) from each dataset.

Dataset	C0	C1	C2	C3	C4	C5	C6	C7	C8	C8	Mean error
AUC	21.5 %	24.5%	16%	29%	19.5%	24.5%	41%	48%	41.5%	23.5%	**28.9%**
DMD	29.5%	22%	28.5%	18.5%	21%	16%	13%	25.5%	23.5%	30.5%	**22.8%**

TABLE 2. Benchmark data sources. V. = Visual, VCD = Vehicle Control Data, PD = Physiological data, R = Realistic, S = Simulated, * = not mentioned.

Reference	Type	Samples	Environment	Distraction types	Advantage	Limitations
[3]	V	34K	R + S	4 normal driving tasks and 3 distracted activities	- Balanced - High frame rate	- Limited number of classes - Not public
[4]	V	35K	S	1 Safe driving and 9 distracted activities	- Critical situations are investigated without risk	- Unrealistic driving behavior - Not public
[5]	V	24k	R+S	1 Safe driving and 3 distracted activities	- Balanced - Rich	- Not public - Different dangerous driving activities fall into the same category
[6]	V	*	*	1 Safe driving and 1 risky	- Useful for simple driver alert systems	- Limited number of classes - Not public
[7]	V	106k	S	1 safe driving and 5 manual distraction activities	- Rich in terms of driver gender and age	- Limited number of classes - Many distraction activities are considered Safe - Unrealistic driving behavior - Not public
[8]	VCD	508k	S	1 Safe and 2 distraction activities	- High resolution - Cognitive distraction scenarios are well established	- Data is not representative (only yonger drivers) - Limited number of classes - Unrealistic driving behavior
[9]	VCD	790k	S	1 Safe and 2 distraction activities	- High resolution	- Limited number of classes - Unrealistic driving behavior - Not public
[10]	V + PD	540k	S	1 Safe and 2 cognitive distraction activities	- Combine PD and V data - Provide interesting features on cognitive distraction	- Limited number of classses - Unrealistic driving behavior - Data is not representative - Not public
[11]	V	22k	R	1 Safe driving and 9 distracted activities	- Rich in terms of driver gender and driving activities - Balanced	- For competition purposes only
[12]	V	17k	R	1 Safe driving and 9 distracted activities	- Rich - Public	- Unbalanced - Annotation issues
[13]	V	199k	R	13 driver actions	- Rich - Public	- Unbalanced - Annotation issues
[14]	V	*	R	1 Safe and 1 visual distraction	- Driver's facial features are clearly visible	- Not public - Limited number of classses

In 2020, a new Driver Monitoring Dataset (DMD) was presented in [13]. All recordings were made from three in-vehicle perspectives with three cameras positioned to capture the driver's face, hands, and body. Each camera offers three channels: RGB, infrared, and depth information. A total of 37 volunteers participated in the creation of DMD, 27% of whom were women.

The authors of the DMD and AUC datasets used a temporal annotation mechanism to annotate the recorded videos. Each video sequence (group of frames) was labeled with one class. However, by analyzing the annotated frames, we noticed that in some cases, the authors failed to correctly select the beginning and the ending of a homogeneous video sequence, which caused incorrect annotations of some of the images. Since the datasets contain a very large number of images, we have randomly selected and examined 2000 images from each dataset (i.e., 200 images from each class) to estimate the rates of incorrect annotations. We found that 28.9% and 22.8% of the labels of AUC and DMD, respectively, are incorrect, as illustrated in Table 1. This is an issue when training neural networks to classify each frame.

Descriptions of the benchmark data sources are given in Table 2.

B. DETECTION METHODS

There are three main methods used to detect driver distraction, as described next:

- Thresholding: this is a training-less method for distraction detection, in which extracted features' values are compared with preset thresholds. For example, in [14], the authors developed a real-time approach to detecting driver distraction by comparing visual features from the face region with thresholds.
- Classical machine learning: this approach detects driver distraction by building a machine learning model whose inputs are engineered features. For example, in [15], the authors focused on visual and cognitive distraction by monitoring the driver's momentary state; visual distraction detection was based on eye gaze and head direction to generate an attention mapping algorithm; cognitive distraction detection relied on eyes, head movements, and vehicle driving position. Rule-based and support

vector machine (SVM) methods were used for the classification task. The results show over 80% success in detecting visual distraction and 68% success in detecting cognitive distraction. In [16], the authors focused on driver cognitive distraction and used head orientation and eye-tracking measurement as well as the interval between the heart R-waves as features for the pattern recognition algorithm (AdaBoost, SVM). In [17], the authors compared the performance of Random Forest and some other well-known classifiers for visual distraction detection. First, five visual features were extracted: arm position, eye closure, eye gaze, facial expressions, and orientation. The extracted features were then fed into a classifier, such as AdaBoost, Hidden Markov Models, Random Forest, SVM, Conditional Random Field, or Neural Network. Experimental results show the superiority of the Random Forest classifier compared to the other classifiers.

- Deep learning: this approach is often referred to as end-to-end learning since feature engineering is not required and models are learned from the raw data. Deep neural networks have achieved outstanding performance on various machine learning tasks, especially computer vision (image recognition, object detection, and semantic segmentation). Driving distraction detection based on image/video analysis has thus benefited from this approach. For example, in [5], the authors utilized the pretrained 19-layer VGG-19 network to extract visual features and recognize driving behavior. In [4], the authors compared and evaluated four deep convolutional neural networks, including VGG-16, AlexNet, GoogleNet, and residual networks for distraction recognition. Experiments were conducted using a driving simulator to evaluate the trained models. The results indicate that GoogleNet outperforms the other models. In [3], the authors first applied the background subtraction (GMM) algorithm to the raw RGB images to extract the driver's body and remove the background. The result of this pre-processing is then fed to a convolutional neural network (CNN) model for distraction classification. It was shown that the background subtraction step significantly improves the classification accuracy. In our work, we too perform image segmentation prior to learning a classifier. However, instead of using a classical segmentation method (e.g., GMM) to detect the driver's body (foreground), we perform a deep learning-based segmentation to extract *each part* of the human body. As shown later, this reduces the complexity of the subsequent classification task and increases its robustness to the noise by selecting only the parts of the body that are relevant for distraction detection. Indeed, in GMM, only moving objects in the scene are considered foreground and thus used for classification. The issue with this approach is that since the camera is installed inside the vehicle, some parts of the driver's body (e.g., head, torso) may not move, and may thus be considered as background, while the driver is engaging in a distraction. This motivates the need for robust driver body detection. The authors of [18] have proposed a driver behavior analysis system using a two-stream convolutional neural network (CNN) model. A two-dimensional (2D) ConvNet was used to construct the spatial and temporal ConvNet streams, which was pre-trained using the ImageNet dataset. Then, a fusion network was designed to integrate the features for classification. The authors achieved an average accuracy of 80% using their own dataset which has not been made public. The authors of [19] tested different machine learning and deep learning approaches to detect cognitive load while driving. The authors exploited visual, thermal, and physiological modalities to model distracted driving behavior and investigated how different modality-fusion and machine-learning processing pipelines could handle various modalities. Experimental results showed that gradient boosting achieved the highest F1 classification score of 94%. In [20], the authors have proposed a spatial-temporal framework that combines CNN and Recurrent Neural Network-based Gated Recurrent Unit to process EEG signals. The framework was evaluated for binary classification tasks (safe and distracted) and achieved an accuracy of 92%. A comparison of the detection methods is given in Table 3.

C. SEGMENTATION METHODS

Image segmentation can be formulated as a problem of classifying pixels with semantic labels (semantic segmentation) or partitioning of individual objects (instance segmentation). The latter extends the semantic segmentation scope further by detecting and delineating each object in the image. Mask R-CNN is one of the robust deep learning-based instance segmentation methods [21]. It extends Faster R-CNN [22] by adding a branch for predicting segmentation masks on each region of interest, parallel with the existing branch for classification and bounding box regression. In [23], the authors proposed a new instance-based human segmentation method that separates instances based on the human pose rather than proposal region detection. Experimental results showed that the proposed method outperforms the Mask R-CNN method on the human instance segmentation problem and better handles the occlusion challenge.

Human body parts segmentation represents a challenging problem in the image segmentation field. It aims at partitioning persons in the image into multiple semantically consistent regions (e.g., head, arms, legs). In [24], the authors proposed the HBPS method, which is a deep learning-based segmentation method with encoding and decoding parts. The encoding part comprises multiple convolutions and pooling layers, whereas the decoding part consists of up-convolution layers. Each layer combines the output of its previous layer with the contracting network's corresponding layer's pooled features. The authors of [25] contributed to the enrichment of the Pascal Visual Object Classes dataset [26] without human labeling by synthetic images. Moreover, they

TABLE 3. Methods comparison. + = fusion of methods.

Reference	Approach	Features	Method	Output
[14]	Thresholding	Visual features from the face region	Thresholding	Visual distraction
[15]	ML	Eye gaze, head direction, vehicle driving position	Rule-based + SVM	Cognitive distraction
[16]	ML	head orientation, eye gaze, heart R-waves	AdaBoost + SVM	Visual and cognitive distraction
[17]	ML	arm position, eye closure, eye gaze, facial expressions, orientation	Random Forest classifier	Visual distraction
[5]	DL	RGB image	VGG-19 network (CNN)	Visual distraction
[4]	DL	RGB image	GoogleNet (CNN)	Manual, visual and cognitive distraction
[3]	DL	RGB image	GMM + AlexNet (CNN)	Manual, visual and cognitive distraction
[18]	DL	Sequence of 10 images	Two stream CNN	Manual, visual and cognitive distraction
[19]	DL	Visual, thermal and physiological features	Gradient Boosting	Cognitive distraction
[20]	DL	(EEG) signals	CNN + Gated Recurrent Unit	Manual, visual and cognitive distraction

proposed the CDCL method, where the model learns parts segmentation from graphics simulation. Experimental results showed that the CDCL outperforms several state-of-the-art approaches requiring human labeling including those proposed in [27]–[32], and [33].

III. DATASET

In our study, due to the limitations mentioned in Table 2, we built our own dataset using an efficient data collection and annotation strategy. The following ten driver distraction classes were defined:

- C0: Safe driving
- C1: Texting - right hand
- C2: Talking on the phone - right hand
- C3: Texting - left hand
- C4: Talking on the phone - left hand
- C5: Operating the radio
- C6: Drinking
- C7: Reaching behind
- C8: Tidying up hair or applying makeup
- C9: Talking to passenger

To study the driver's behavior in real traffic situations, we conducted experiments using an instrumented vehicle, which comprises: (i) a camera, installed above the vehicle's side window and oriented toward the driver, and (ii) a Mobile Digital Video Recorder (MDVR).

One part of the data was collected in real-world driving conditions. The other part was collected by asking drivers to simulate different types of driving behaviors in the instrumented vehicle, but without moving the vehicle for safety reasons. Nine drivers were involved in the experiment. Each of them was asked to perform the ten activities separately (i.e., one activity for each video sequence) while driving or pretending to drive, which took about 15 minutes for each driver resulting in about 450 images per class per driver. After manual examination, a total of about 38 thousand images were kept in the dataset. Table 4 shows the data distribution over the classes.

TABLE 4. Class distribution of the collected dataset.

Class	Number of samples	%
C0	3879	10.1
C1	3792	9.9
C2	3959	10.3
C3	3807	9.9
C4	4018	10.5
C5	3853	10.1
C6	3629	9.5
C7	3849	10
C8	3791	9.9
C9	3750	9.8

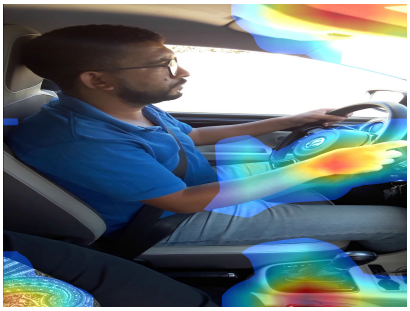


FIGURE 1. Discriminative pixels used by the VGG-19 model trained on the raw RGB images to activate the safe driving class.

IV. METHODOLOGY

The main component of the proposed framework is the human body parts segmentation. It is applied to the raw RGB image in order to efficiently remove the irrelevant objects and identify the driver's critical body parts. The resulting image is then fed into the classification model; see fig. 2. In this section, we first discuss the human body parts segmentation step, and then we describe the classification models and training methods used to classify the segmented image.

A. SEGMENTATION PROCESS

The collected images contain many useless regions (e.g. the vehicle cabin with its various components, and the external environment), which disturb the building of an accurate CNN-based distraction detection classifier. fig. 1 shows the discriminative pixels used by the VGG-19 classifier when trained on raw RGB images to activate the safe driving class (more details are given in V-B). As safe driving is defined by keeping the eyes on the road and the hands on the steering wheel, the crucial regions that the CNN classifier must use are those corresponding to the hands and the head only. However, as shown in fig. 1, the pixels that contributed to activate

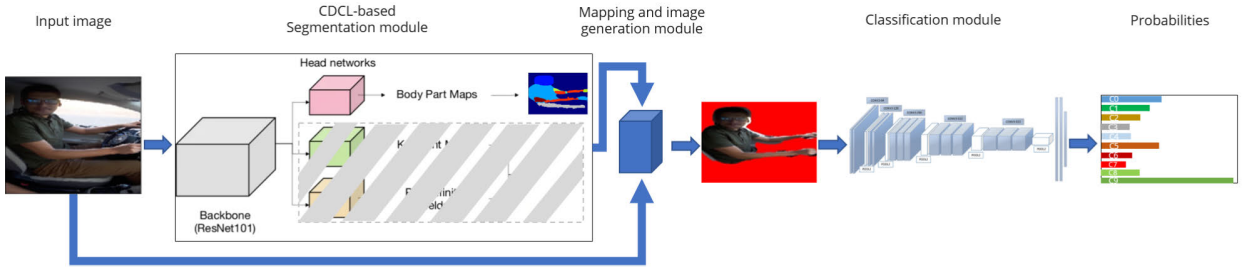


FIGURE 2. General system architecture.

the safe driving class include also other pixels which are irrelevant to the distraction classification. Indeed, the roof of the vehicle and the gearbox does not provide any information about the driver's distraction. A solution to this problem is to force the CNN classifier to focus on the critical pixels, i.e. those contributing to the driver's distraction recognition.

The simplest method towards this objective is to crop the image to keep only the driver's region according to a predefined Region Of Interest (ROI). However, due to the driver's different activities and the various positions of the driver's seat, defining a good ROI is not possible. A better solution is to segment the driver's body.

Segmentation methods can be classified into two main categories: motion-based segmentation methods and appearance-based segmentation methods. The former cannot segment static parts of the driver's body (i.e., head, torso, etc.) Hence, in our work, we focus on the latter.

We have investigated two segmentation architectures: Human Body Parts Segmentation (i.e. HBPS) [24] and Cross-Domain Complementary Learning (i.e. CDCL) [25].

Both segmentation methods detect each part of the human body (torso, arms, legs, head, etc.) and assign an integer to each of the predefined human parts in the Pascal VOC dataset. They achieve great performance on the Pascal VOC dataset; mean Intersection Over Union (mIOU) = 70% for CDCL and mIOU = 71% for HBPS. However, the setting used to collect our data differs from that of the Pascal VOC dataset. Indeed, as mentioned above, in our setting, the camera is installed above the vehicle's side window, and thus the camera captures the side view of the driver. Therefore, we need to reevaluate the performance of the two methods on our dataset to choose the more efficient one.

To assess the performance of the HBPS and CDCL models on our dataset, knowledge of the associated models' architectures and weights is required. This knowledge is available for the CDCL model (the weights are published in GitHub [34]). However, the authors of HBPS did not publish the model's weights. So, we have trained the HBPS architecture on the Pascal VOC dataset. We used the early stopping technique to avoid the overfitting problem, and we let the model run for 1000 epochs. The model achieved a validation log-loss of 0.34 and a validation accuracy of 88% (see Table 5).

TABLE 5. HBPS training results. TLL: Train log-loss, VLL: Validation log-loss, TACC: Train accuracy, VACC: Validation accuracy.

Epoch	TLL	VLL	TACC	VACC
1	2.60783	1.14133	0.77695	0.01420
100	0.45808	0.47295	0.87658	0.86603
200	0.40335	0.43511	0.89062	0.87358
300	0.38004	0.41896	0.89619	0.87721
400	0.36017	0.40883	0.9013	0.87933
500	0.34115	0.3973	0.90556	0.88069
600	0.32121	0.38287	0.90977	0.8838
700	0.30312	0.36712	0.91319	0.88677
800	0.28986	0.35936	0.91583	0.88754
900	0.27603	0.35337	0.91848	0.88766
1000	0.26975	0.3476	0.91923	0.88844

We then tested both models (i.e., CDCL and HBPS) on our dataset. The models generate an output mask consisting of an image-map classifying each pixel into one of the body part categories. It is worth pointing out that the CDCL [25] predicts two additional outputs: (1) a set of confidence keypoint maps and (2) a set of Part Affinity Fields that we have ignored in this study as shown in fig. 2.

Although both models achieved almost the same performance on the Pascal VOC dataset, the CDCL model achieved significantly better performance on our dataset (as presented in V-A). Indeed, with the CDCL model, a good segmentation of critical body parts (i.e., hands, head, arms, torso, neck) is obtained even when the image is captured in low lighting conditions (see fig. 3). In this study, by using the mapping and image generation module, the critical driver's body parts obtained by the segmentation module are selected from the original image and fed to the classification module (see fig. 2).

B. CLASSIFICATION MODEL LEARNING

1) CNN MODELS

Two CNN models have been investigated in our work:

- VGG-19 model: this is a pre-trained model that contains 19 layers with very small receptive fields (3 × 3); it was proposed in [35] for large-scale image recognition, and was one of the famous models submitted to the ILSVRC-2014 challenge [35]. It achieved an improvement over AlexNet [36] by replacing large kernel-sized filters (11 and 5 in the first and second convolutional

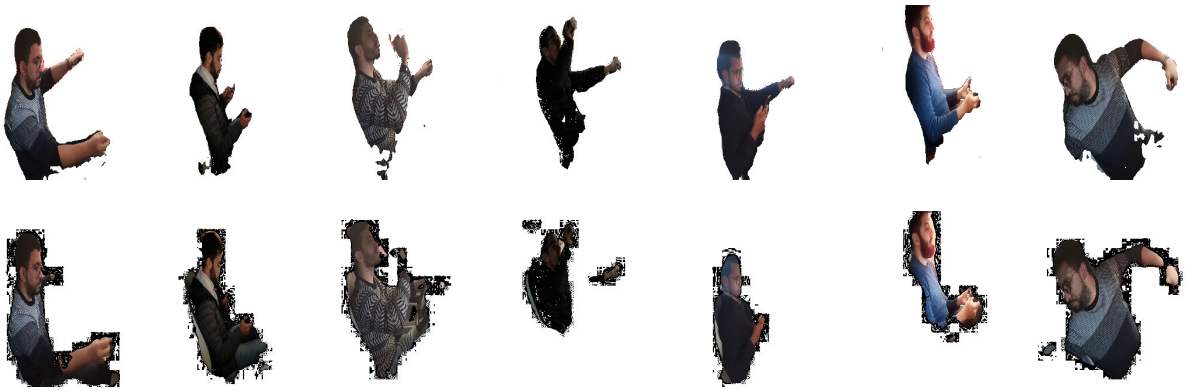


FIGURE 3. Qualitative results of the segmentation methods. The first (resp. second) row illustrates the segmented images using the CDCL (resp. HBPS) model.

layers, respectively) with multiple 3×3-sized filters one after another.

- Inception-v3 model: this architecture, which was proposed in [37], is the winner of the 2015 ILSVRC challenge, with 3.5% top-5 error and 17.3% top-1 error on the validation set, and 3.6% top-5 error on the official test set. This model can be seen as the culmination of many ideas which were developed by multiple researchers over the years. The model is made up of symmetric and asymmetric building blocks, including convolutions, average pooling, max pooling, concats, dropouts, and fully connected layers. Batch norm is used extensively throughout the model and applied to activation inputs. The loss is computed via the Softmax function.

2) TRAINING METHODS

The two commonly used approaches for training a deep neural network are: i) transfer learning plus fine tuning of the last layers, and ii) training the neural network from scratch. Both approaches have benefits and can be used for different deep learning tasks.

Transfer learning is a widely used technique, which involves reusing a pre-trained model, which was trained on a large benchmark dataset (i.e. Imagnet dataset) to solve a similar problem. The last fully-connected (FC) layers are modified to adjust to the downstream task's requirements.

For our driving classification task, the last fully connected layer computes the probability associated with each of the ten classes.

Transfer learning works well if the images used to train the last fully connected layers and to test the trained model are somewhat closely related to the images used to pre-train the neural network, i.e. the ImageNet dataset. However, in many real-world applications of CNN for image recognition, the distribution of the images to be classified may be significantly different from those of ImageNet images. This is particularly true for the detection of driver distraction. To mitigate this issue, one can either unfreeze some pre-trained layers and train them along with the fully connected layers, or train the entire neural network model from scratch. In this work we have opted for the former approach.

Instead of random initialization, we initialize the weights of the VGG-19 and Inception-v3 models using the ImageNet-based pre-trained weights, and we unfreeze some layers in order to retrain them using our dataset. The multi-perceptrons classifier is modified as described in Table 8). The other aspects of the training method are standard operations in deep learning.

V. EXPERIMENTAL RESULTS AND ANALYSIS

In this section, we evaluate and analyze the performance of the proposed driving distraction detection framework, on the proposed dataset and on a correctly labeled subset of AUC benchmark. The main aspect to consider is the evaluation of the impact of the segmentation module on classification performance.

A. PERFORMANCE OF THE PROPOSED FRAMEWORK

To evaluate the performance of the proposed framework, we first started by segmenting the collected dataset. For each segmented image, we kept only the driver's body parts relevant to the classification task (as mentioned in IV-A). Then, we divided the segmented images into training and test sets. Since our goal is to design a general solution that works well on any vehicle, we used the leave-one-out (LOO) cross-validation method to get the distraction classification results for each driver. The images from one driver are used as testing images, whereas the rest of the images associated with the other eight drivers are used for training. Therefore, for each driver, the data composing the testing set is entirely new to the trained CNN models.

The following techniques were applied during the training process to avoid overfitting on the training dataset and improve the recognition results.

- Data augmentation: the images were cropped and flipped horizontally to expand the size of the training dataset.

TABLE 6. Range of hyper-parameters used for optimization.

Hyper-parameter	Values/Range
Learning rate	$[1 \times 10^{-2}, 1 \times 10^{-3}, 1 \times 10^{-4}, 1 \times 10^{-5}]$
dropout	[0.2-0.7]
Nbr of FC layers	[2-4]
Nbr of neurons	[512-4096]

TABLE 7. Hyper-parameters optimization results for the classification models.

Hyper-parameter	VGG-19*	Inception-v3
Learning rate	1×10^{-5}	1×10^{-3}
Nbr of FC layers	2	2
Nbr of Neurons	4096	4096
Dropout ratio	0.5	0.5

- Dropout [38]: this was applied to each of the fully connected layers except for the last one.
- Adam optimizer: this was used to explore the parameter space better and thus minimize the cost function as much as possible.
- Keras Tuner optimizer with Randomsearch: this was used in our work to find the optimal learning rate for the Adam optimizer, the optimal dropout ratio for each of the dropout layers, and the best number of fully connected (FC) layers with the best number of neurons. This was developed by the Google team and is included in the Keras open library [39]. First, we split the training set into an optimization training set and an optimization validation set (the optimization validation set represents 10% of the training set). Then, the Keras tuner optimizer performs an iteration loop, which evaluates a number of combinations of hyper-parameter values. The evaluation is performed by calculating the precision of the trained model on the validation set.
  The range of hyper-parameter values used in the optimization procedure are listed in Table 6:

The optimal values for the hyper-parameters of the VGG-19 and Inception-v3 classifiers are found to be the same for the nine different training sets. Table 7 represents the best Hyper-parameters obtained for the two CNN models. As can be seen, the optimal number of fully connected layers of the VGG-19 model is reduced by one compared to the original architecture [35], (i.e., the number of layers is changed from 19 to 18). We denote this modified architecture by VGG-19*. Table 8 describes the architecture of the VGG-19* and the Inception-v3 models.

As shown in Table 9, the general detection accuracy for the CDCL-based segmentation with the Inception-v3 model achieved an average of 92.04%. The bottom row represents the weighted average detection rate for each activity, which is defined by:

$$\text{waACC}_k = \frac{\sum_{i=1}^{n} w_{ik} \text{ACC}_{ik}}{\sum_{i=1}^{n} w_{ik}} \quad (1)$$

where waACC_k is the weighted average accuracy for the kth activity, n is the number of drivers, ACC_{ik} is the detection accuracy for the ith driver's kth activity, and w_{ik} is the number of images of the ith driver belonging to the kth activity. The rightmost column of Table 9 represents the average detection results for each driver.

Table 10 illustrates the CDCL-based distraction detection results when using the VGG-19* model. The average accuracy of this method is 96.25%.

The HBPS-based distraction detection was similarly investigated using VGG-19* and Inception-v3 models and achieved 77,78% and 76,89%, respectively, as illustrated in Tables 11 and 12. The decrease in precision compared to the classification based on CDCL can be explained by the cumulative error resulting from the HBPS model.

In order to better analyze the benefits and the drawbacks of our approach, a confusion matrix is computed to report the results of the studied multi-classification problem. Fig. 4 shows the confusion matrix of the proposed approach. It reports the classification results of the nine experiments (i.e., each time we conduct the experiment on a new driver using LOO). The green diagonal elements show the number of correct classifications for each class. The last column shows the classification recall (i.e., fraction of the total amount of relevant instances that were actually retrieved), and the last row shows the classification precision (i.e. the fraction of relevant instances among the retrieved instances).

By inspecting the confusion matrix, we can see that all of the driving activities were detected with a good accuracy except for the 8th activity (i.e. tidying up hair and applying makeup), 150 cases of which were misclassified as 'talking on the phone with the right-hand,' in addition to 104 other cases misclassified as 'talking on the phone with the left-hand.' One explanation for this is that the hand gestures and the head orientation might be similar for these classes, thus making the discrimination between them a challenging task for the classifier.

B. ABLATION STUDY

Since our approach used segmented images to classify driver distraction, one interesting question is whether the segmentation model is sound. To answer this question, we evaluated the performance of the classification without the segmentation module. We repeat the same steps as in V-A. We used the raw RGB images instead of the segmented images. We achieved an average accuracy of 77.09% and 75.24% for the VGG-19* and Inception-v3 models, respectively. This shows that the improvement in accuracy brought by the segmentation module exceeds 20% as illustrated in Tables 13 and 14.

In order to understand why CNN models perform so well on the segmented images, the class activation maps (CAM) [40] have been implemented and analyzed. This is a simple technique to identify the discriminative image regions used by a CNN model to identify a specific class in the image. In other words, CAM help us to visualize where CNN pay attention. We select randomly 10 images representing 10 activities from one driver, which have never been seen by the training process. The raw and segmented

TABLE 8. Architectures of the VGG-19 and Inception-v3 models.

	VGG-19*	Inception-v3
Input	image(224,224,3)	image(299,299,3)
Convolutional part	Conv 3 × 3 Conv 3 × 3 max pooling layer Conv 3 × 3 Conv 3 × 3 max pooling layer 4 × Conv 3 × 3 max pooling layer 4 × Conv 3 × 3 max pooling layer 4 × Conv 3 × 3 max pooling layer	Conv 3 × 3 Conv 3 × 3 Conv 3 × 3 max pooling layer Conv 1 × 1 Conv 3 × 3 max pooling layer 3 × Inception 5 × Inception 2 × Inception
multilayer perceptron classifier	Fully connected layer - 4096 Dropout layer Fully connected layer - 10	GlobalAveragePooling2D Fully connected layer - 4096 Dropout layer Fully connected layer 10

TABLE 9. Accuracy results for driving distraction classification based on CDCL and Inception-v3.

Drivers	Distraction activities										Average accuracy
	C0	C1	C2	C3	C4	C5	C6	C7	C8	C9	
D1	86.52	87.59	86.58	96.66	94.19	91.40	64.22	92	94.81	98	89.19
D2	100	87	85.92	100	99.28	100	98.84	99.05	85.47	100	95.55
D3	96.15	98.89	98.19	94.62	95.97	94.40	91.29	100	96.44	98.48	96.44
D4	97.15	85.14	98.89	100	98.44	100	64	96.74	84.65	99.78	92.47
D5	99.12	99.56	65.86	99.56	93.62	100	82.30	96.26	86.50	94.18	88.52
D6	100	96.79	77.7	96.64	100	91.26	100	100	93.83	99.28	95.55
D7	92.98	87.31	86.59	82.18	93.64	97.39	65.59	99.57	93.82	89.01	88.80
D8	100	100	87.08	96.01	75	99.56	82.30	100	65.70	90	89.56
D9	99.66	74.29	100	71.24	100	89.67	84.53	74.53	81.14	100	87.5
Weighted average accuracy	96.64	90.52	87.46	93.13	94.27	95.96	81.07	97.83	87.30	96.26	**92.04**

TABLE 10. Accuracy results for driving distraction classification based on CDCL and VGG-19*.

Drivers	Distraction activities										Average accuracy
	C0	C1	C2	C3	C4	C5	C6	C7	C8	C9	
D1	99.78	96.46	92.37	99.16	99.81	100	100	83.56	98.70	99.26	96.91
D2	99.76	100	84.77	100	98.56	99.77	88.70	100	79.03	100	95.05
D3	98.72	98.89	92.15	94.84	97.09	90.20	100	98.87	95.33	100	96.60
D4	92.76	100	88.89	100	81.78	100	100	85.82	86.91	95.80	93.19
D5	100	100	99.56	100	98.30	100	85.84	100	81.64	99.57	96.49
D6	97.37	94.64	85.21	87.22	100	98.65	99.69	99.70	98.24	97.12	95.78
D7	98.68	100	98.70	100	98.05	100	99.34	100	81.14	87.93	96.38
D8	100	97.53	87.75	100	99.34	99.34	100	100	86.84	98.67	96.94
D9	100	99.78	98.24	100	98.12	100	98.83	99.56	98.68	100	99.32
Weighted average accuracy	98.50	98.79	92.25	97.82	96.79	98.83	96.91	95.97	89.26	97.49	**96.25**

TABLE 11. Accuracy results for driving distraction classification based on HBPS and Inception-v3.

Drivers	Distraction activities										Average accuracy
	C0	C1	C2	C3	C4	C5	C6	C7	C8	C9	
D1	77.01	67.78	73.65	70.11	81.23	80.12	77.27	78.61	72.66	80.14	75.86
D2	80.11	65.78	72.25	71.32	79.08	79.88	79.68	77.54	71.48	80.25	75.74
D3	79.13	70.76	69.88	70.4	80.68	79.64	79.87	73.22	71.67	83.61	75.89
D4	79.61	65.22	74.81	69.89	79.79	81.32	77.67	79.36	72.14	84.23	76.40
D5	82.78	70.28	74.23	73.84	80.73	81.68	79.55	78.82	72.67	82.64	77.72
D6	78.11	68.41	72.43	71.23	82.68	79.37	78.81	77.37	75.18	86.33	76.99
D7	84.61	70.09	70.67	73.84	80.48	80.22	80.64	76.76	76.76	85.83	77.99
D8	83.44	71.34	71.59	69.46	79.89	79.98	78.31	78.13	78.35	86.76	77.73
D9	81.36	73.65	74.58	70.63	78.11	81.67	78.35	75.38	73.49	85.76	77.30
Weighted average accuracy	80.67	69.36	72.79	71.23	80.36	80.40	78.96	77.36	73.89	83.88	**76.89**

images are fed into the prediction process, and the CAM are computed to generate a heat map that shows the strongest activations.

Fig. 5 shows the activation maps when applying the classification model to the images shown in the top row of this figure. The Middle row represents the discriminate

TABLE 12. Accuracy results for driving distraction classification based on HBPS and VGG-19*.

Drivers	Distraction activities										Average accuracy
	C0	C1	C2	C3	C4	C5	C6	C7	C8	C9	
D1	70.23	78.15	74.11	72.46	83.22	79.87	81.66	79.87	75.22	87.22	78.20
D2	69.55	76.32	73.25	71.35	81.99	81.34	82.46	78.26	74.61	84.69	77.38
D3	72.16	70.54	72.68	72.83	85.15	81.06	82.63	75.34	75.68	85.11	77.32
D4	64.42	69.90	74.68	70.33	81.27	83.75	79.11	81.22	73.24	86.89	76.48
D5	73	67.53	76.87	73.29	78.76	82.71	82.61	80.71	70.86	88.42	77.48
D6	74.36	69.11	75.12	72.04	82.22	83.13	79.39	79.62	78.51	87.37	78.09
D7	76.22	76.90	73.45	73.24	84.69	79.69	81.34	78.89	79.61	85.27	78.93
D8	63.28	74.62	72.21	71.26	84.76	78.07	81.22	80.99	78.73	86.78	77.19
D9	76.3	73.95	74.69	72.81	83.43	83.91	82.86	79.61	75.28	87.51	79.04
Weighted average accuracy	70.83	73.06	74.17	72.18	82.82	81.35	81.50	79.50	75.81	86.56	**77.78**

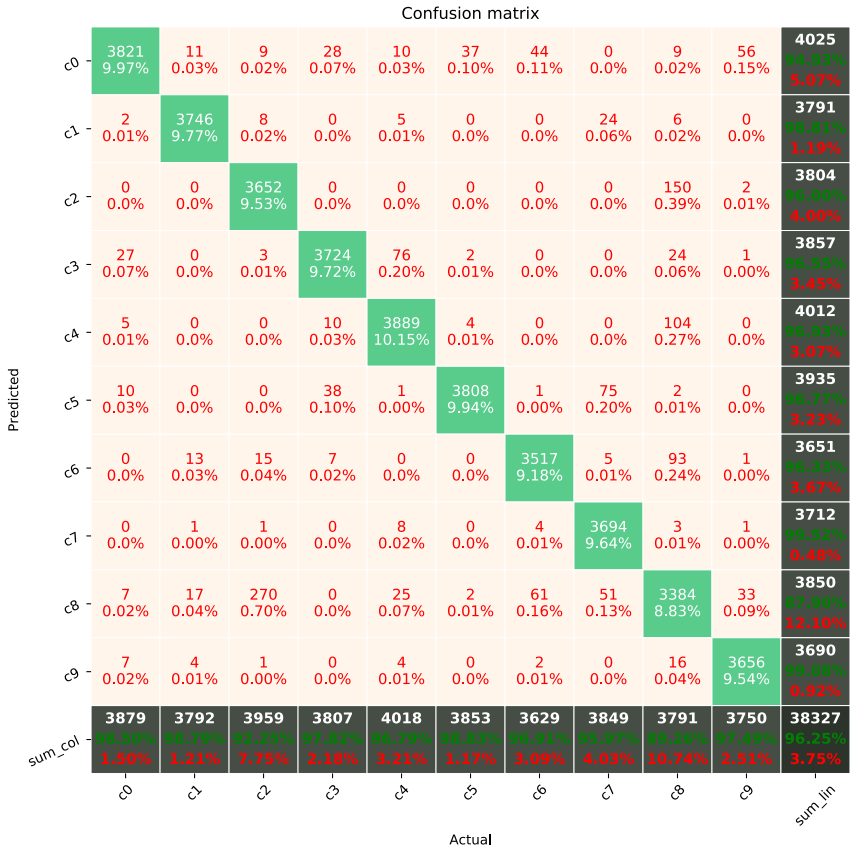


FIGURE 4. Confusion matrix of the distraction detection results.

TABLE 13. Accuracy results for driving distraction classification using VGG-19* on raw RGB images.

Drivers	Distraction activities										Average accuracy
	C0	C1	C2	C3	C4	C5	C6	C7	C8	C9	
D1	79.12	65.53	72.91	69.01	81.78	80.54	79.3	77.06	73.16	86.54	76.50
D2	81.23	66.98	72.82	69.74	81.66	80.11	80.02	75.45	71.66	85.99	76.57
D3	79.47	70.63	70.89	70.11	82.78	79.79	81.46	71.68	72.87	84.67	76.44
D4	80.87	65.54	73.15	68.22	79.33	82.44	75.66	79.65	69.81	86.78	76.15
D5	84.09	68.61	74.44	68.64	79.89	81.66	79.43	78.82	68.78	87.12	77.15
D6	78.21	69.11	72.88	71.65	81.06	82.79	78.84	77.02	74.73	86.88	77.32
D7	86.11	71.22	70.33	70.15	82.84	78.13	80.19	76.99	75.19	86.46	77.76
D8	83.12	70.45	73.18	69.82	83.03	77.99	79.66	78.15	77.48	87.01	77.99
D9	82.71	72.76	72.13	68.55	81.99	81.67	80.87	76.65	72.99	86.79	77.71
Weighted average accuracy	81.64	69.06	72.56	69.57	81.59	80.51	79.50	77.00	73.00	86.48	**77.09**

regions given by the ReLU16 layer of the VGG-19* model (i.e., last convolutional layer) when applied to the raw images (no segmentation). The bottom images represent the discriminate regions given by the ReLU16 layer of the VGG-19* model when applied to images generated from the segmentation module.

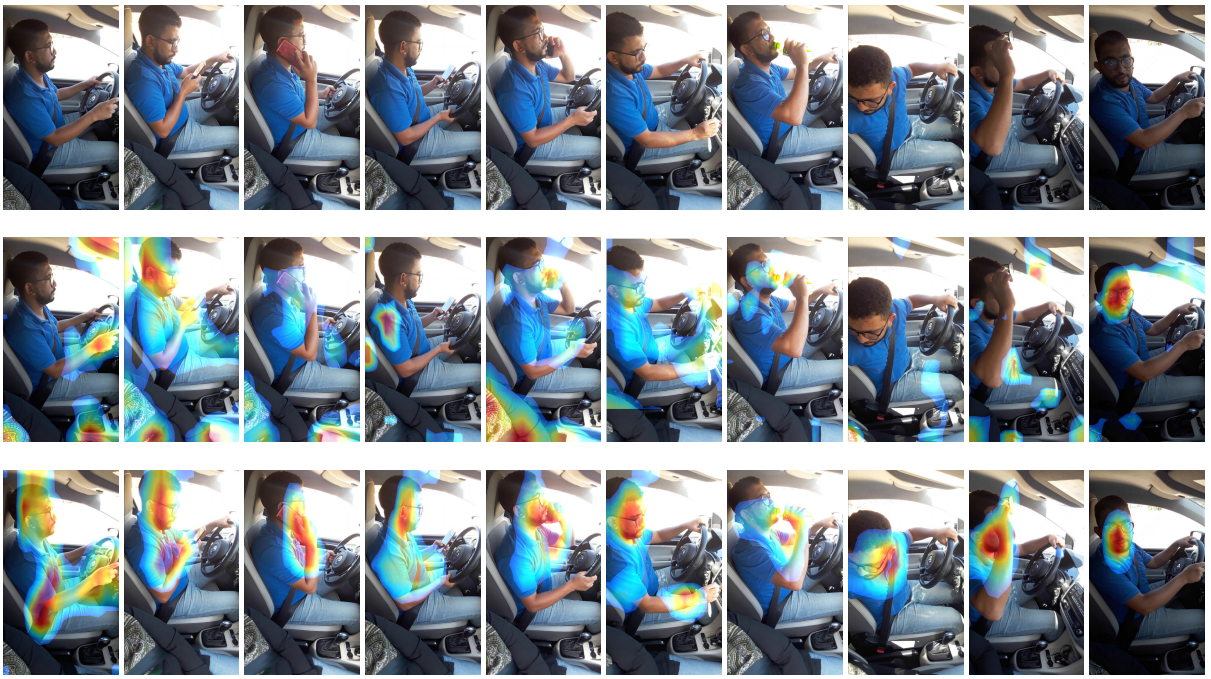


FIGURE 5. Discriminative image regions used by a CNN to identify a specific class in the image. The first row illustrates the raw RGB images, the second and last rows represent the discriminative image regions of the VGG-19* model without segmentation and the proposed model, respectively.

TABLE 14. Accuracy results for driving distraction classification using Inception-v3 on raw RGB images.

Drivers	Distraction activities										Average accuracy
	C0	C1	C2	C3	C4	C5	C6	C7	C8	C9	
D1	76.44	66.67	71.22	66.82	79.56	77.68	74.54	76.32	71.87	79.87	74.10
D2	79.65	64.32	70.68	67.61	78.64	78.14	78.26	75.84	69.59	80.16	74.79
D3	78.62	69.05	69.13	68.86	80.14	77.63	79.47	69.78	70.76	82.46	74.59
D4	78.14	63.68	72.66	67.58	77.66	79.78	76.87	77.67	69.86	83.67	75.46
D5	82.31	68.11	72.76	66.34	77.86	77.59	77.62	76.04	70.22	81.35	75.02
D6	77.89	66.87	70.34	70.34	81.22	76.62	76.38	76.86	71.68	85.98	75.42
D7	84.67	68.23	69.88	67.46	79.66	77.24	79.32	74.69	74.87	85.78	76.18
D8	82.65	69.44	67.94	65.87	78.53	76.86	77.86	77.63	76.21	86.76	75.98
D9	80.76	71.84	70.08	66.03	77.68	79.47	77.49	72.97	71.65	85.43	75.34
Weighted average accuracy	80.11	69.03	70.51	67.47	79.02	77.82	77.68	75.40	71.92	83.42	**75.24**

As shown in the first row of fig. 3, the CDCL-based segmentation module followed by the mapping and image generation module (see fig. 2) can extract the critical parts of the driver body effectively, which allows the classification module to focus on the most relevant features for distraction detection. This is illustrated in the bottom row of images of fig. 5, where the strongest activations of the proposed approach are associated with the driver's head rotation and the arm's position. The strongest activations for the raw image-based classifier (i.e., classification model without segmentation) includes, in addition to the driver's head rotation and the arm's position, unnecessary information that are considered as noise, (e.g. rear mirror position, gear-stick position, etc. CNN models trained on segmented images are shown to successfully identify and localize the discriminative regions for driver distraction detection.

C. COMPARISON WITH OTHER METHODS

We furthermore compared the proposed approach in the public benchmark AUC with three state-of-the-art approaches, including [4], [18], and [41]. The authors of [18] performed a re-split of the AUC dataset, letting the methods train on a set of drivers and tested on never seen drivers. As the proposed framework was trained on our dataset, we randomly selected a set of 2000 images, corrected the corresponding annotations, and then carried out the evaluation. Similarly, we re-implemented the solutions proposed in [4], trained the Resnet and GoogleNet networks on our dataset, and then tested them on the benchmark subset. The obtained results are shown in Table 15.

TABLE 15. Classification results comparison on AUC dataset.

Method	Dataset	Accuracy
Spatial Stream ConvNet [18]	re-split of AUC	76.25%
VGG-16 with regularization [41]	re-split of AUC	77.15%
Resnet [4]	subset of AUC	76.12%
GoogleNet [4]	subset of AUC	75.68%
Ours with Inception-v3	subset of AUC	93.42%
Ours with VGG-19*	subset of AUC	**95.77%**

D. DRIVER DISTRACTION DETECTION: BINARY CLASSIFICATION

Road accidents involving distracted drivers can be avoided if the driver is alerted when distraction is detected. In a simplified setting, the driver assistance system may warn the driver only if he/she is distracted regardless of the type of distraction and without attributing any risk level to the detected distraction. In this setting, we convert the multi-class classification problem into a binary classification problem. In order not to retrain the models, we combine/aggregate classification results from the multi-class classifier to get the average accuracy for the binary classification problem. Hence, we select one-ninth of the samples from each distracted driving class set, and then merge them all into a single distraction set in order to obtain a balanced data set. We ran the proposed framework over the created data set. Then, the multi-classification results are aggregated to calculate the precision of the binary classifier. The accuracy of the binary classifier achieves 99%, thus making the proposed method an efficient solution for distracted driving warning systems.

VI. CONCLUSION

In this work, we proposed a solution to detect driver distraction based on first segmenting the raw images to detect the critical driver's body parts, and then apply a deep Convolutional Neural Network (CNN), which was trained using transfer learning and fine-tuning using a dataset that we have built using an instrumented vehicle and nine drivers. Extensive experimental results have shown that the segmentation module significantly improves the classification performance, with an average accuracy exceeding 96%.

In the future, we will enrich our dataset with images of drivers of diverse ages and ethnic groups. We are also currently building an embedded system to implement a driving warning system based on the proposed driver distraction detection solution.

APPENDIX

The data that support the findings of this study are openly available at github.com/AmalEzzouhri/Driver-Distraction-Dataset

REFERENCES

[1] National Center for Statistics and Analysis, "Distracted driving," Nat. Highway Traffic Saf. Admin., Washington, DC, USA, Res. Rep. DOT HS 812 926, Apr. 2010.

[2] National Safety Council. (2021). *Ending Distracted Driving is Everyone's Responsibility*. [Online]. Available: https://www.nsc.org/road-safety/safety-topics/distracted-driving

[3] 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.

[4] D. Tran, H. M. 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, 2018.

[5] A. Koesdwiady, S. M. Bedawi, C. Ou, and F. Karray, "End-to-end deep learning for driver distraction recognition," in *Proc. Int. Conf. Image Anal. Recognit.* Cham, Switzerland: Springer, Jul. 2017, pp. 11–18.

[6] O. Wathiq and B. D. Ambudkar, "Optimized driver safety through driver fatigue detection methods," in *Int. Conf. Trends Electron. Inform. (ICEI)*, May 2017, pp. 68–73.

[7] L. Li, B. Zhong, J. C. Hutmacher, Y. Liang, W. J. Horrey, and X. Xu, "Detection of driver manual distraction via image-based hand and ear recognition," *Accident Anal. Prevention*, vol. 137, Apr. 2020, Art. no. 105432.

[8] M. Atiquzzaman, "Exploring distracted driver detection algorithms using a driving simulator study," Ph.D. dissertation, Southern Illinois Univ. Edwardsville, Edwardsville, IL, USA, 2016.

[9] L. Jin, Q. Niu, H. Hou, H. Xian, Y. Wang, and D. Shi, "Driver cognitive distraction detection using driving performance measures," *Discrete Dyn. Nature Soc.*, vol. 2012, Oct. 2012, Art. no. 432634.

[10] H. Kawanaka, M. Miyaji, M. S. Bhuiyan, and K. Oguri, "Identification of cognitive distraction using physiological features for adaptive driving safety supporting system," *Int. J. Veh. Technol.*, vol. 2013, pp. 1–18, Jul. 2013.

[11] S. Kaggle. (2016). *State Farm Distracted Driver Detection*. [Online]. Available: https://www.kaggle.com/c/state-farm-distracted-driver-detection

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

[13] J. D. Ortega, N. Kose, P. Cañas, M. A. Chao, A. Unnervik, M. Nieto, and L. Salgado, "DMD: A large-scale multi-modal driver monitoring dataset for attention and alertness analysis," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, Aug. 2020, pp. 387–405.

[14] L. Alam and M. M. Hoque, "Real-time distraction detection based on driver's visual features," in *Proc. Int. Conf. Electr., Comput. Commun. Eng. (ECCE)*, Feb. 2019, pp. 1–6.

[15] M. Kutila, M. Jokela, G. Markkula, and M. R. Rue, "Driver distraction detection with a camera vision system," in *Proc. IEEE Int. Conf. Image Process.*, Jan. 2007, pp. 201–211.

[16] M. Miyaji, M. Danno, H. Kawanaka, and K. Oguri, "Driver's cognitive distraction detection using AdaBoost on pattern recognition basis," in *Proc. IEEE Int. Conf. Veh. Electron. Saf.*, Sep. 2008, pp. 51–56.

[17] A. Ragab, C. Craye, M. S. Kamel, and F. Karray, "A visual-based driver distraction recognition and detection using random forest," in *Proc. Int. Conf. Image Anal. Recognit.* Cham, Switzerland: Springer, Oct. 214, pp. 256–265.

[18] J. C. Chen, C. Y. Lee, P. Y. Huang, and C. R. Lin, "Driver behavior analysis via two-stream deep convolutional neural network," *Appl. Sci.*, vol. 10, no. 6, p. 1908, 2020.

[19] M. Gjoreski, M. Z. Gams, M. Lustrek, P. Genc, J. U. Garbas, and T. Hassan, "Machine learning and end-to-end deep learning for monitoring driver distractions from physiological and visual signals," *IEEE Access*, vol. 8, pp. 70590–70603, 2020.

[20] G. Li, W. Yan, S. Li, X. Qu, W. Chu, and D. Cao, "A temporal-spatial deep learning approach for driver distraction detection based on EEG signals," *IEEE Trans. Automat. Sci. Eng.*, early access, Jun. 24, 2021, doi: 10.1109/TASE.2021.3088897.

[21] K. He, G. Gkioxari, P. Dollar, and R. Girshick, "Mask R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2961–2969.

[22] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 91–99.

[23] S.-H. Zhang, R. Li, X. Dong, P. Rosin, Z. Cai, X. Han, D. Yang, H. Huang, and S.-M. Hu, "Pose2Seg: Detection free human instance segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 889–898.

[24] G. L. Oliveira, A. Valada, C. Bollen, W. Burgard, and T. Brox, "Deep learning for human part discovery in images," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2016, pp. 1634–1641.

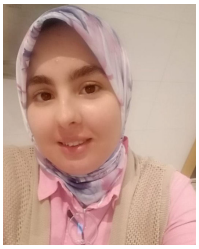
[25] K. Lin, L. Wang, K. Luo, Y. Chen, Z. Liu, and M. T. Sun, "Cross-domain complementary learning using pose for multi-person part segmentation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 3, pp. 1066–1078, Dec. 2020.

[26] The PASCAL Visual Object Classes. *Pascal VOC Dataset*. Accessed: Dec. 20, 2021. [Online]. Available: http://host.robots.ox.ac.uk/pascal/VOC/

[27] F. Xia, P. Wang, L. C. Chen, and A. L. Yuille, "Zoom better to see clearer: Human and object parsing with hierarchical auto-zoom net," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, Oct. 2016, pp. 648–663.

[28] L.-C. Chen, Y. Yang, J. Wang, W. Xu, and A. L. Yuille, "Attention to scale: Scale-aware semantic image segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 3640–3649.

[29] X. Liang, X. Shen, D. Xiang, J. Feng, L. Lin, and S. Yan, "Semantic object parsing with local-global long short-term memory," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 3185–3193.
[30] K. Gong, X. Liang, D. Zhang, X. Shen, and L. Lin, "Look into person: Self-supervised structure-sensitive learning and a new benchmark for human parsing," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 932–940.
[31] X. Liang, X. Shen, J. Feng, L. Lin, and S. Yan, "Semantic object parsing with graph LSTM," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, Oct. 2016, pp. 125–143.
[32] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 834–848, Apr. 2017.
[33] H.-S. Fang, G. Lu, X. Fang, J. Xie, Y.-W. Tai, and C. Lu, "Weakly and semi supervised human body part parsing via pose-guided knowledge transfer," 2018, *arXiv:1805.04310*.
[34] *CDCL Human Part Segmentation*. Accessed: Dec. 20, 2021. [Online]. Available: https://github.com/kevinlin311tw/CDCL-human-part-segmentation
[35] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*.
[36] A. Krizhevsky and I. S. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.* 2012, pp. 1097–1105.
[37] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2818–2826.
[38] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, Jan. 2014.
[39] Keras. (2021). *KerasTuner*. [Online]. Available: https://keras-team.github.io/keras-tuner/
[40] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2921–2929.
[41] 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.



AMAL EZZOUHRI received the B.S. degree in mathematics and computer science, and the M.Sc. degree in data science from the Faculty of Science Dhar Mahraz, Sidi Mohamed Ben Abdellah University, Fez, Morocco, in 2016 and 2018, respectively. She is currently pursuing the Ph.D. degree with Mohammed V University, Rabat, Morocco, and the International University of Rabat, Morocco. Her research interests include machine learning and its application to intelligent transportation systems and road security.

ZAKARIA CHAROUH received the B.S. degree in mathematics and computer science from the University of Ibn Zohr, Agadir, Morocco, in 2014, and the M.Sc. degree in information systems security from the National School of Applied Sciences of Kenitra, Morocco, in 2016. He is currently pursuing the Ph.D. degree in data science with Mohammed V University, Rabat, Morocco, and the International University of Rabat. He is also an Adjunct Professor of computer sciences with the College of Engineering, International University of Rabat. His research interests include computer vision, artificial intelligence, and intelligent transportation systems.

MOUNIR GHOGHO (Fellow, IEEE) received the M.Sc. and Ph.D. degrees from the National Polytechnic Institute of Toulouse, France, in 1993 and 1997, respectively. He was an EPSRC Research Fellow with the University of Strathclyde, Scotland, from September 1997 to November 2001. In December 2001, he joined the School of Electronic and Electrical Engineering, University of Leeds, England, where he was promoted to a Full Professor, in 2008. While still affiliated with the University of Leeds, in 2010, he joined the International University of Rabat, Morocco, where he is currently the Dean of the Doctoral College and the Director of the ICT Research Laboratory (TIC Lab). He is also the Co-Founder and the Co-Director of the CNRS-Associated International Research Laboratory DataNet. He held invited Scientist/Professor positions at Telecom Paris-Tech, France; NII, Japan; BUPT, China; the University Carlos Third of Madrid, Spain; ENSICA, Toulouse; Darmstadt Technical University, Germany; and Minnesota University, USA. His research interests include signal processing, machine learning, wireless communication, and their applications. He was elevated to the grade of IEEE Fellow in 2018. He was awarded the U.K. Royal Academy of Engineering Research Fellowship in 2000 and the IBM Faculty Award in 2013. In the past, he served as an Associate Editor for the *IEEE Signal Processing Magazine*, the IEEE TRANSACTIONS ON SIGNAL PROCESSING, the IEEE SIGNAL PROCESSING LETTERS, and the *Digital Signal Processing* journal (Elsevier); and a member of the IEEE Signal Processing Society SPCOM Technical Committee, the IEEE Signal Processing Society SPTM Technical Committee, and the IEEE Signal Processing Society SAM Technical Committee. He is also a member of the Steering Committee of the IEEE TRANSACTIONS ON SIGNAL AND INFORMATION PROCESSING OVER NETWORKS. He organized many conferences and workshops, including the 2019 Intelligent Environments Conference, the 2018 ITCities Workshop, the 2013 European Signal Processing Conference (EUSIPCO), and the 2010 IEEE Workshop on Signal Processing for Advanced Wireless Communications (SPAWC).

ZOUHAIR GUENNOUN (Senior Member, IEEE) received the B.E. degree in electronics and telecommunications from the Electronics and Electrical Montefiore Institute, ULG Liege, Belgium, in 1987, and the M.Sc. degree in communication systems and the Ph.D. degree from the EMI School of Engineering, Rabat, Morocco, in 1993 and 1996, respectively. He visited the Centre for Communication Research (CCR), Bristol University, U.K., from 1990 to 1994, to prepare a split Ph.D. From 1988 to 1996, he worked as an Assistant Lecturer with the EMI School of Engineering, where he has been a Professor Lecturer, since 1996. He is currently in charge of the Research Laboratory in Electronics and Telecommunications (LEC) at EMI. His research interests include digital signal processing, error control coding, and speech and image processing. He was a member of the Moroccan IEEE Section Executive Committee.