

Research Article

Driver Distraction Identification with an Ensemble of Convolutional Neural Networks

Hesham M. Eraqi ¹, Yehya Abouelnaga ²,
Mohamed H. Saad ³, and Mohamed N. Moustafa ¹

¹Department of Computer Science & Engineering, The American University in Cairo, Egypt

²Department of Informatics, Technical University of Munich, Germany

³Department of Computer and Systems Engineering, Ain Shams University, Egypt

Correspondence should be addressed to Hesham M. Eraqi; heraqi@aucegypt.edu
and Yehya Abouelnaga; yehya.abouelnaga@tum.de

Received 6 October 2018; Revised 27 December 2018; Accepted 14 January 2019; Published 13 February 2019

Guest Editor: Yasser Hassan

Copyright © 2019 Hesham M. Eraqi et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The World Health Organization (WHO) reported 1.25 million deaths yearly due to road traffic accidents worldwide and the number has been continuously increasing over the last few years. Nearly fifth of these accidents are caused by distracted drivers. Existing work of distracted driver detection is concerned with a small set of distractions (mostly, cell phone usage). Unreliable *ad hoc* methods are often used. In this paper, we present the first publicly available dataset for driver distraction identification with more distraction postures than existing alternatives. In addition, we propose a reliable deep learning-based solution that achieves a 90% accuracy. The system consists of a genetically weighted ensemble of convolutional neural networks; we show that a weighted ensemble of classifiers using a genetic algorithm yields a better classification confidence. We also study the effect of different visual elements in distraction detection by means of face and hand localizations, and skin segmentation. Finally, we present a thinned version of our ensemble that could achieve 84.64% classification accuracy and operate in a real-time environment.

1. Introduction

Over the 20 years from 1980 to 2000, the number of licensed drivers in the United States increased by 23.7%, to reach 190.6 million licenses [1]. From 1990 to 2000, the urban vehicles miles traveled increased by 80%, while roads building rose by only 37% [2]; this makes driving a common activity for many people and makes the driving safety an important issue in everyday life. Construction of new roads did not keep up with the noticeable increase in vehicles, leading to more traffic congestion [2]. In addition, In-Vehicle Information Systems (IVISs) such as media players and navigation devices introduce more distraction to the driving experience and lead to more accidents.

Despite safety improvements in road and vehicle design, the total number of fatal crashes still increases [1]. The 2017 Global Status Report of the World Health Organization (WHO) reported an estimated 1.25 million yearly deaths due

to road traffic accidents worldwide, with up to 50 million people sustaining non-fatal injuries as a result of road traffic accidents [3]. To put that mortality rate into perspective, it is the same as having around seven Boeing airplanes crashing or disappearing every day. Car accidents mortality rate compares to those of serious diseases (i.e., Hepatitis & HIV). At the same year, the numbers of global deaths attributable to Hepatitis and HIV are estimated to be in the order of 1.3 million [4] and 1.1 million [3], respectively, which is almost the same as the number of the people dying yearly due to road traffic accidents. Moreover, road traffic accidents cause a huge property damage, and the number of road accidents due to distracted driving is steadily increasing.

Nearly fifth of traffic accidents are caused by a distracted driver according to the National Highway Traffic Safety Administration (NHTSA), with approximately 90% of road accidents being due to human errors in the United States [5]. Despite the fact that vehicle crashes are attributed to multiple

causes, driver error represents a dominant factor [6]. In 2015, 3,477 people were killed, and 391,000 were injured in motor vehicle crashes involving distracted drivers [7]. The major cause of these accidents was the use of mobile phones [7].

The NHTSA defines distracted driving as “any activity that diverts attention from driving”, including (a) talking or texting on one’s phone, (b) eating and drinking, (c) talking to passengers, or (d) fiddling with the stereo, entertainment, or navigation system [7]. The Center for Disease Control and Prevention (CDC) provides a broader definition of distracted driving by taking into account visual (i.e., taking one’s eyes off the road), manual (i.e., taking one’s hands off the driving wheel), and cognitive (i.e., taking one’s mind off driving) causes [8].

One way to address the problem of distracted driver is to develop distraction mitigation systems, which adapt IVIS functions according to driver state. In such a mitigation system, correctly identifying driver distraction is critical, which is the focus of this work. We envision a future where smart vehicles could detect and identify such distraction and warn the driver against it or take preventive measures. On the other hand, such detection systems can help law enforcement to identify distraction on highway using radar cameras and penalize certain forms of distraction. Moreover, the recent commercial semiautonomous cars require drivers to pay attention to the road [9]. Autonomous steering control [10] systems require drivers to be ready to take back control of the wheel [9]. This is what makes distracted drivers detection an important system component in these cars. Distraction detection can also be used to enable Advanced Driver Assistance Systems (ADAS) features [11] like Collision Avoidance Systems (CAS) that have to plan evasive maneuvers [12].

Research in the field of distracted driving detection follows the definitions presented in [7, 8]. It detects manual, visual, or cognitive types of distractions. Cognitive distractions deal with tasks of listening, conversing, daydreaming, or just becoming lost in thought. In this form of distraction, the driver is “mentally” distracted from safe driving even though they are in a safe driving posture. Visual distractions often refer to situations where the driver takes their eyes off the road due to either “the presence of salient visual information away from the road causing spontaneous off-road eye glances and momentary rotation of the head” or the use of multimedia devices (i.e., cell phones, navigation, or entertainment systems) [13]. Visual distractions are coined in the following terms: “sleepiness”, “drowsiness”, “fatigue”, and “inattention”. And, they usually depend on facial landmarks detection and tracking. Manual distractions are mainly concerned with driver’s activities other than safe driving (i.e., reaching behind, adjusting hair and makeup, or eating and drinking). In this kind of distraction, authors often tend to depend heavily on hand tracking and driving posture estimation. In this paper, we focus only on “manual” distractions where a driver is distracted by texting or using cell phone, calling, eating or drinking, reaching behind, fiddling with the radio, adjusting hair and makeup, or talking to a passenger.

Recently, the field witnessed a dramatic increase in computational power (thanks to massive parallelization in modern GPUs) and available big data for deep learning

training. That enabled more research on advanced/deep network architectures. One example is the proliferation of deep learning-based solutions utilizing convolutional neural networks to solve computer vision problems, which achieved unprecedented performance [15–17]. That improvement is a byproduct of learning feature maps (as in [15, 18]) rather than hand-crafting them in traditional computer vision practices [19].

In this paper, we detect and identify distraction using deep learning. RGB images are obtained from a camera mounted above the dashboard. We train and benchmark multiple convolutional neural network architectures. We use pretrained networks in a “transfer learning”-style where networks are pretrained on the ImageNet dataset [17] (that is, 1.2 million images and 1000 classes). This initializes the network with a generic set of features and allows it to recognize a variety of objects with high precision. Then, we retrain the latent fully connected layers to recognize the distraction postures (instead of ImageNet classes). During training, the convolutional layers are usually either kept frozen or updated with a miniscule learning rate. In such way, we transfer the networks knowledge of about 1,000 different objects (i.e., the broad domain) to our more specific domain of 10 driving distraction postures.

We present a real-time system for driver distraction identification that uses a learnable weighted ensemble of convolutional neural networks (CNNs), a new method for skin segmentation, a challenging distracted driver’s dataset (see Figure 1) on which we evaluate our proposed solution, and an annotation tool [20] for action labeling that can be used to extend our dataset.

2. Literature Review

The work in the distracted driver detection field over the past seven years could be clustered into four groups: multiple independent cell phone usage detection publications, Laboratory of Intelligent and Safe Automobiles in University of California San Diego (UCSD) datasets and publications, Southeast University Distracted Driver dataset and affiliated publications, and recently StateFarm’s Distracted Driver Kaggle competition.

2.1. Cell Phone Usage Detection. Reference [21] presents an SVM-based model that detects the use of mobile phone while driving (i.e., distracted driving). Their dataset consists of frontal image view of a driver’s face. They also make premade assumptions about hand and face locations in the picture. Reference [22] presents another SVM-based classification method to detect cell phone usage. However, their dataset is collected from transportation imaging cameras that are deployed in highways and traffic lights, which is, indeed, more competitive. Reference [23] uses AdaBoost classifier and Hidden Markov Models to classify a Kinect’s RGB-D data. Their solution depends on indoor-produced data. They sit on a chair and a mimic a certain distraction (i.e., talking on the phone). This setup misses two essential points: the lighting conditions and the distance between a Kinect and



FIGURE 1: Ten classes of driver postures from our dataset.

the driver. In real-life applications, a driver is exposed to a variety of lighting conditions (i.e., sunlight and shadow). Reference [24] suggests using a Hidden Conditional Random Fields (HCRF) model to detect cell phone usage. Their model operates face, mouth, and hand features of images obtained from a camera mounted above the dashboard. Reference [25] devised a Faster-RCNN model to detect driver's cell phone usage and "hands on the wheel". Their model is mainly geared towards face/hand segmentation. They train their Faster-RCNN on the dataset proposed in [26] (that we also use in this paper). Their proposed solution runs at a 0.06 and 0.09 frames per second for cell phone usage and "hands on the wheel" detection. Reference [27] tackles the problem of cell phone usage detection. Their approach does not hold any static assumptions though (i.e., in which region of the image a face is expected to be found). They use a Supervised Descent Method (SDM) to localize the face landmarks and, then, extract two bounding boxes to the left and the right side of the face. They train a classifier on each of the two regions to detect cell phone usage: right hand, left hand, or no usage. Using a histogram of gradients (HOG) and an AdaBoost classifier, they achieve a 93.9% classification accuracy and operate in a near real-time speed (7.5 frames per second).

2.2. UCSD's Laboratory of Intelligent and Safe Automobiles Work. Reference [28] presents a vision-based analysis framework that recognizes in-vehicle activities using two Kinect cameras that provide frontal and back views of the driver. Their approach provides "hands on the wheel" information (i.e., left hand only, both hands, no hands) and uses this information to detect three types of distractions: adjusting the radio, operating the gear, and adjusting the mirrors. Reference [29] presents a fusion of classifiers where the image is to be segmented into three regions: wheel, gear, and instrument panel (i.e., radio). It proposes a classifier for each segment to detect existence of hands in those regions. The hand information (i.e., output of the classifiers) is passed to an "activity classifier" that infers the actual activity (i.e., adjusting the radio, operating the gear). Reference [30] extends existing research to include eye cues to previously existing head and hands cues. However, it still considers three types of distractions: "wheel region interaction with two hands on the wheel, gear region activity, and instrument cluster region activity". Reference [31] presents a region-based classification

approach. It detects hands presence in certain predefined regions in an image. A model is learned for each region separately. All regions are later joined using a second-stage classifier.

2.3. Southeast University Distracted Driver Dataset. Reference [32] designs a more inclusive distracted driving dataset with a side view of the driver and more activities: grasping the steering wheel, operating the shift lever, eating a cake, and talking on a cellular phone. It introduces a contourlet transform for feature extraction and, then, evaluates the performance of different classifiers: Random Forests (RF), k -Nearest Neighbors (KNN), and Multilayer Perceptron (MLP). The random forests achieved the highest classification accuracy of 90.5%. Reference [33] showed that using a multiwavelet transform improves the accuracy of Multilayer Perceptron classifier to 90.61% (previously 37.06% in [32]). Reference [34] showed that using a Support Vector Machine (SVM) with an intersection kernel, followed by Radial Basis Function (RBF) kernel, achieved the highest accuracies of 92.81% and 94.25%, respectively (in comparison with [32, 33]). After testing against other classification methods, they concluded that an SVM with intersection kernel offers the best real-time quality (67 frames per second) and better classification performance. Reference [35] improves the Multilayer Perceptron classifier using combined features of Pyramid Histogram of Oriented Gradients (PHOG) and spatial scale feature extractors. Their Multilayer Perceptron achieves a 94.75% classification accuracy. Reference [36] utilizes motion history images (HMI) to make use of the data's temporality. Pyramid Histogram of Gradients (PHOG) is applied to the motion history images. A Random Forrest trains on the extracted features and yields a 96.56% accuracy. Reference [37] presents a convolutional neural network solution that achieves a 99.78% classification accuracy. They train their network in a 2-step process. First, they use pretrained sparse filters as the parameters of the first convolutional layer. Second, they fine-tune the network on the actual dataset. Their accuracy is measured against the 4 classes of the Southeast dataset.

2.4. StateFarm's Dataset. StateFarm's Distracted Driver Detection competition on Kaggle was the first publicly available dataset for posture classification. In the competition,

StateFarm defined ten postures to be detected: 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 passenger. Our work, in this paper, is mainly inspired by StateFarm’s Distracted Driver’s competition. While the usage of StateFarm’s dataset is limited to the purposes of the competition [38], we designed a similar dataset that follows the same postures.

3. Dataset Design

Creating a new dataset was essential to the completion of this work. The available alternatives to our dataset are StateFarm and Southeast University (SEU) datasets. StateFarm’s dataset is to be used for their Kaggle competition purposes only (as per their regulations) [38]. As for Southeast University (SEU) dataset, it presents only four distraction postures. And, after multiple attempts to obtain it, we figured out that the authors do not make it publicly available. All the papers ([32–37, 39]) that benchmarked against the dataset are affiliated with either Southeast University, Xi’an Jiaotong-Liverpool University, or Liverpool University, and they have at least one shared author. With that being said, the collected “distracted driver” dataset is the first publicly available (obtainable after signing a license agreement) for driving posture estimation research. Our dataset is publicly available, subject to signing our agreement form from [40]. The dataset introduced in this work is an extended and cleaned-up version of our dataset presented in [41].

3.1. Camera Setup. Our dataset collection setup has a single camera with a fixed perspective, and the data collection was conducted on two phases. In each phase, a different camera is used. In one phase we use the ASUS ZenFone smartphone (Model ZD551KL) rear camera [42], and in the other phase we used the DS325 Sony DepthSense camera [43]. The latter camera provides depth information, but we only record the RGB images. Collecting data from different cameras adds an extra dimension of diversity to our dataset, and we demonstrate the feasibility of effective distraction detection by relying on RGB cameras which are widely available and of low cost.

The data was collected in a video format and, then, cut into individual images, 1080×1920 or 640×480 each. The cameras are fixed using an arm strap to the car roof handle on top of the front passenger’s seat. In our use case, this setup proved to be very flexible as we needed to collect data in different vehicles.

3.2. Labeling. In order to label the collected videos, we designed a simple multiplatform action annotation tool using modern web technologies: Electron, AngularJS, and JavaScript. The annotation tool is open-source and publicly available at [20].

3.3. Statistics. We had 44 participants from 7 different countries: Egypt (37), Germany (2), USA (1), Canada (1), Uganda

(1), Palestine (1), and Morocco (1). Out of all participants, 29 were males and 15 were females. Some drivers participated in more than one recording session with different time of day, driving conditions, and wearing different clothes. Videos were shot in 5 different cars: Proton Gen 2, Mitsubishi Lancer, Nissan Sunny, KIA Carens, and a prototyping car. We extracted 14,478 frames distributed over the following classes: safe driving (2,986), phone right (1,256), phone left (1,320), text right (1,718), text left (1,124), adjusting radio (1,123), drinking (1,076), hair or makeup (1,044), reaching behind (1,034), and talking to passenger (1,797). The sampling is done manually by inspecting the video files with eye and giving a distraction label for each frame. The transitional actions between each consecutive distraction types are manually removed. Figure 1 shows samples for the ten classes in our dataset.

4. Proposed Method

Our proposed solution consists of a genetically weighted ensemble of convolutional neural networks. The convolutional neural networks are trained on raw images, skin-segmented images, face images, hands images, and “face+hands” images. On those five images sources, we train and benchmark an AlexNet network [15], an InceptionV3 network [44], a ResNet network having 50 layers [45], and a VGG-16 network [46]. We fine-tune a pretrained ImageNet model (i.e., transfer learning) for these networks. Then, we evaluate a weighted sum of all networks’ outputs yielding the final class distribution using a genetic algorithm. The system overview is shown in Figure 2.

4.1. Skin Segmentation. Skin segmentation is a challenging problem to solve, mainly due to the different lighting conditions happening during driving. We use a Multivariate Gaussian Naive Bayes classifier to develop a pixel-wise skin segmentation model. Our model is similar to [47] except that we do not use a histogram as a Likelihood function. Instead, we fit the training data into Gaussian distributions to formulate the Likelihood functions. The posterior probability is evaluated as in (1).

$$\mathbb{P}(\text{skin} | x) = \frac{\mathbb{P}(x | \text{skin}) \cdot \mathbb{P}(\text{skin})}{\mathbb{P}(x)}, \quad (1)$$

$$\mathbb{P}(x | \text{skin}) \sim \mathcal{N}(\mu_{\text{skin}}, \Sigma_{\text{skin}})$$

Note that $\mathbb{P}(\text{skin}) = \mathbb{P}(\text{non-skin}) = 0.5$ (i.e., we do not make any assumptions about existence of skin pixels in the image).

We trained our model using the UCI Skin Segmentation dataset [48]. The database contains RGB colors that are labeled for the skin and non-skin classes. It is generated using skin textures from face images of people from diverse ages, genders, and races. It contains a total of 245,057 color samples, out of which 50,859 are the skin samples and 194,198 are non-skin samples. Two Gaussian distributions (Likelihoods) are constructed for the skin and the non-skin classes by estimating $\mu_{\text{skin}}, \Sigma_{\text{skin}}, \mu_{\text{non-skin}},$ and $\Sigma_{\text{non-skin}}$ from the training data. For deployment phase, each pixel x in

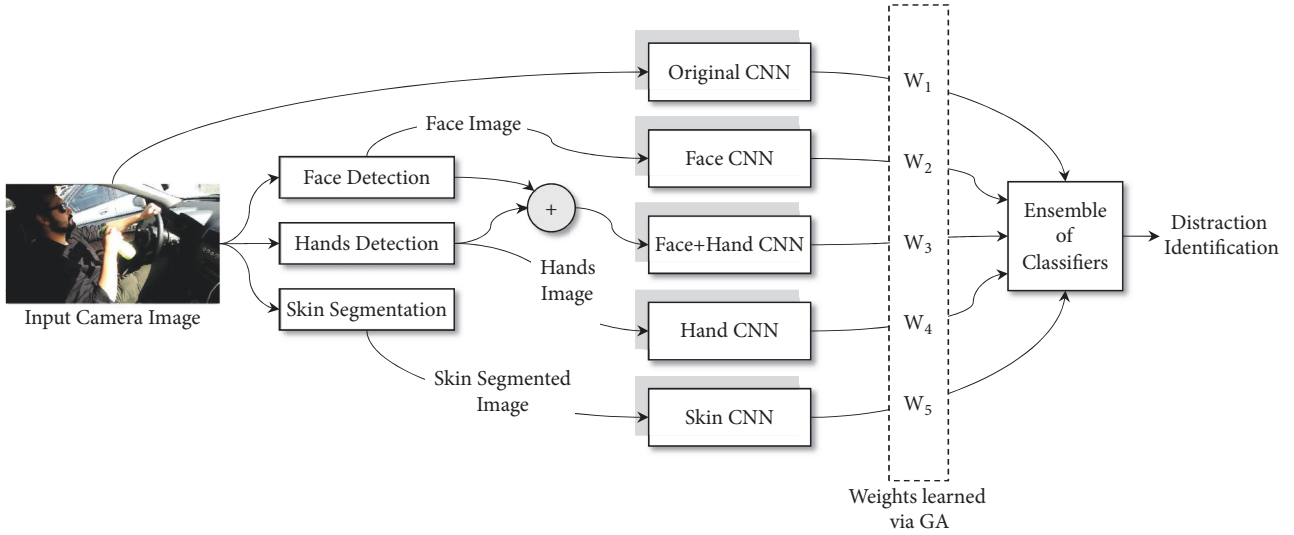


FIGURE 2: An overview of our proposed solution. A face detector, a hand detector, and a skin segmenter are run against each frame. For each output image (i.e., Skin, Face, Hands), an AlexNet and an InceptionV3 network are trained (i.e., resulting in 10 neural networks: 5 AlexNet and 5 InceptionV3). The overall class distribution is determined by the weighted sum of all softmax layers. The weights are learned using a genetic algorithm.



FIGURE 3: Left and right images show a sample for skin segmentation result without and with training the pixels spatial information, respectively.

the input image is fed to the model as in (2). And then, a probability heatmap of skin in the image can be constructed. We classify a pixel to a “skin” if $\text{Model}(x) > 0.5$. Then, we cluster the skin pixels into objects and remove those with a small number of pixels, because neither faces nor hands skin blobs are expected to have small number of pixels.

$$\begin{aligned} \text{Model}(x) &= \frac{\mathbb{P}(\text{skin} | x)}{\mathbb{P}(\text{skin} | x) + \mathbb{P}(\text{non-skin} | x)} \\ &= \frac{\mathbb{P}(x | \text{skin})}{\mathbb{P}(x | \text{skin}) + \mathbb{P}(x | \text{non-skin})} \end{aligned} \quad (2)$$

One key disadvantage of such method is that it is very sensitive to image illumination conditions. Hence, incorporating pixel location can improve the skin classification accuracy. One way is to pass the pixel location to the input feature vector. However, to the best of our knowledge, there is no available dataset to train and evaluate such method. Besides, annotating a new dataset is costly. Therefore, we adopt an active learning-based approach to supervise the training. The above classifier (without pixel spatial information) is run

against all training images to generate skin masks. Generated masks are manually inspected to cherry-pick samples with high skin segmentation accuracy. Those pixels are used as new training data of the proposed skin segmentation classifier, such that the feature vector includes pixels spatial information (X- and Y-coordinates within the images) in addition to the color information (Red, Green, and Blue color components). Figure 3 shows a sample skin-segmented image with (right) and without (left) pixel spatial information. We notice an accuracy improvement after considering the pixels spatial information. More test data results are presented in Section 5.

4.2. Face and Hands Detection. We trained the model presented in [49] on the Annotated Facial Landmarks in the Wild (AFLW) face dataset [50]. It was sensitive to distance from the camera; faces that were close to the camera were not easily detected. We found that the pretrained model (presented in [14]) produced better results on our dataset. Given that we did not have any hand labeled face bounding boxes, we could not formally compare the two models. But it was obvious that [14] gives a better detection accuracy based on inspecting the results manually. However, face misdetections are noticed in

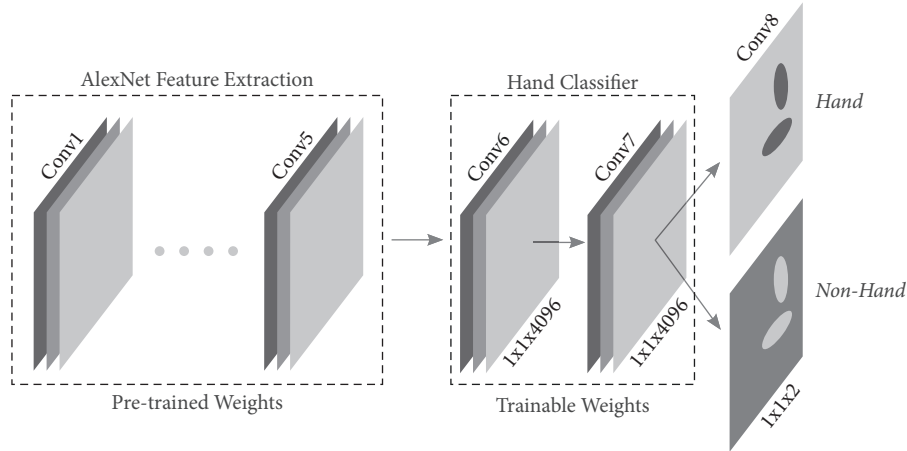


FIGURE 4: A modified version of AlexNet for hands localization (“Conv” stands for a convolutional layer). An AlexNet is trained on “hands” and “non-hands” images. Fully connected layers are replaced with convolutional layers with the same weights of the fully connected layers.

several examples, mainly because the detector is not trained to handle non-frontal faces.

As for hands detection, we used the pretrained model presented in [51] with modifications. Their trained model was a binary class AlexNet that classifies hands/non-hands for different proposal windows. We transferred the weights of the fully connected layers (i.e., fc6, fc7, and fc8) into convolutional layers such that each neuron in the fully connected layer was transferred into a feature map layer with a 1-pixel kernel size. Our proposed architecture, shown in Figure 4, accepts variant-size inputs and produces variant-size outputs. The last convolutional layer has a depth of 2 (i.e., the binary classes), and for each pixel the summation of the two depths is one as in (3), where W and H are the output’s width and height, respectively.

$$\text{Conv8}_{x,y,0} + \text{Conv8}_{x,y,1} = 1, \quad 0 \leq x \leq W \quad 0 \leq y \leq H \quad (3)$$

4.3. Convolutional Neural Network. For distracted driver posture classification, we trained and benchmarked different neural networks architectures: an AlexNet [15], an InceptionV3 [44], a ResNet network having 50 layers [45], and a VGG-16 network [46]. Each network is trained on 5 different image sources (i.e., raw, skin, face, hands, and face+hands images).

We trained our AlexNet models from scratch. We did not use a pretrained model. As for InceptionV3, we performed transfer learning. We fine-tuned a pretrained model [52] on the distraction postures. We removed the “logits” fully connected layer and replaced it with 10-neuron fully connected layer (i.e., corresponding to 10 driving postures). For all of our models, we used a gradient descent optimizer with an initial learning rate of 10^{-2} . The learning rate decays linearly in each epoch with a step of $(10^{-2} - 10^{-4})/\text{Epochs}$. We trained the networks for 30 epochs. In each epoch, we divide the training dataset into minibatches of 50 images each.

4.4. GA-Based Ensemble of Classifiers. Each classifier produces a class probability vector (i.e., output of the “softmax”

layer), $C_1 \dots C_N$, such that $C_i \in \mathbb{R}^{10}$ is a vector having 10 probabilities (for 10 distraction classes) and N is the number of classifiers. In a majority voting system as in (4), it is assumed that all experts (i.e., classifiers) can equally contribute to a better decision by taking the unweighted sum of all classifier outputs.

$$C_{\text{Majority}} = \frac{1}{N} \sum_i^N C_i \quad (4)$$

However, that is not usually a valid assumption. In a weighted voting system as in (5), we assume that classifiers do not contribute equally to the ensemble and that some classifiers might yield higher accuracy than others. Therefore, there is a need to estimate the weights of each classifier’s contribution to the ensemble. Reference [53] presents a variety of methods to estimate the weights. We opted to use a genetic algorithm (i.e., a search-based method).

$$C_{\text{Weighted}} = \frac{1}{\sum_i^N w_i} \sum_i^N w_i \cdot C_i \quad (5)$$

In our genetic algorithm, a chromosome consists of N genes that correspond to the weights $w_1 \dots w_N$. Our fitness function evaluates the Negative Log Likelihood (NLL) loss over a 50% random sample of the population. This helps prevent overfitting. Our population consists of 50 individuals. In each iteration, we retain the top 20% of the population and use them as parents. Then, we randomly select 10% of the remaining 80% of the population as parents. In other words, we have 30% of the population as parents. Now, we randomly mutate 5% of the selected parents. Finally, we cross-over random pairs of the parents to produce children until we have a full population (i.e., with 50 individuals). We ran the above procedure for only 5 iterations in order to avoid overfitting. We selected the chromosome with the highest fitness score (test against all data points, not 50%).

TABLE 1: Distracted driver posture classification results on randomly selected test data.

Model	Source	Loss (NLL)	Accuracy (%)
AlexNet	<i>Original</i>	0.3909	93.65
	Skin Segmented	0.3468	93.62
	Face	1.0516	84.28
	Hands	0.6186	89.52
	Face + Hands	0.8298	86.68
InceptionV3	<i>Original</i>	0.2654	95.17
	Skin Segmented	0.2903	94.66
	Face	0.6096	88.82
	Hands	0.4546	91.62
	Face + Hands	0.4495	90.88
AlexNet		0.2727	94.29
<i>Majority Voting Ensemble</i>		0.1661	95.77
<i>GA-Weighted Ensemble</i>		0.1575	95.98

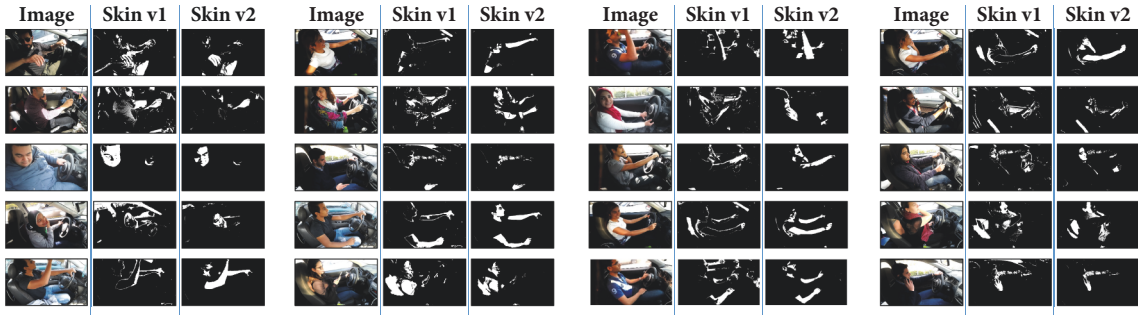


FIGURE 5: Our skin segmentation algorithm (v1 & v2) is run against sample test images. “Skin v2” is trained on pixel spatial information, while “Skin v1” is only trained on pixel color information.

5. Experiments

We divided our dataset into 75% for training and 25% held out as test data using random sampling. Then, we ran the skin segmentation, face and hand detectors on the entire dataset. We tested different deep networks against our test dataset and obtained the results reported in Table 1. We notice that both AlexNet and InceptionV3 achieve best accuracies when trained on the original images. However, the accuracy does not majorly change in both architectures when switching from the original images to skin-segmented images. Hands seem to have more weight in posture recognition than the face. “Face+hands” images produce slightly lower accuracy than the hands images alone, yet still higher than the face images alone. That happens due to face/hand detector failures. For example, if a hand is not found, we pass a face image to a “face+hands” classifier. This does not happen in individual cases of hand-only or face-only classifier because if the hand/face detection fails, we pass the original image to the hand/face classifier as a fallback mechanism. With better hand/face detectors, the “face+hands” networks are expected to produce higher accuracies than the “hands” networks. The confusion matrix of our genetically weighted ensemble of

classifiers on this randomly selected test data is shown in Table 2, and the results are shown in Table 1.

In Figure 5, we show sample results of our skin segmentation algorithm. Test data are never seen by the skin segmentation component during training. In the “Skin v2” column, the classifier is trained on pixels’ spatial information. In the “Skin v1” column, it is only trained on pixels color information. In addition to such qualitative evaluation for our skin detection method, in Table 3 we report a quantitative comparison of distraction classification accuracy by training on skin images generated with and without using pixels’ spatial information. An AlexNet is trained and tested on skin-segmented images.

Table 3 first row result is for a model trained on images generated without spatial information. In the third row, active learning is used to supervise learning pixels’ spatial information with pixels generated from 8,662 cherry-picked skin-segmented images. 100 pixels are randomly sampled from each image to provide an annotated training dataset of 866,200 pixels. A third model is trained on only 50% of these pixels and the result is reported in the second row. The spatial information is proven to improve the skin detection accuracy, while the overall method performance remains

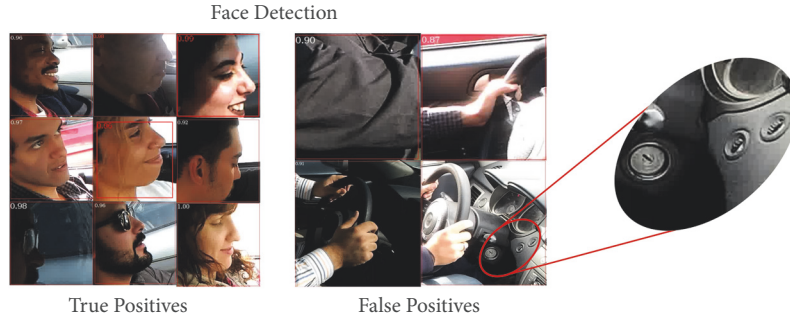


FIGURE 6: Non-frontal faces detection is challenging. A correctly detected face and a wrongly detected one by [14]. Air-conditioning openings are wrongly detected as a face.

TABLE 2: Confusion matrix of our genetically weighted ensemble of classifiers on randomly selected test data.

		Predicted									
		C0	C1	C2	C3	C4	C5	C6	C7	C8	C9
Actual	C0	95.34	0	0.33	0.65	0.11	0.43	0.43	0.87	0.11	1.74
	C1	0.31	96.63	1.23	0.31	0.92	0	0.31	0	0.31	0
	C2	0.29	3.23	96.48	0	0	0	0	0	0	0
	C3	2.02	0.61	0	96.15	0.81	0	0.20	0	0	0.20
	C4	0	0.33	0	4.90	94.77	0	0	0	0	0
	C5	4.26	0	0	0.33	0	95.08	0	0	0	0.33
	C6	0.74	0	0	0.25	0	0.74	98.01	0.25	0	0
	C7	3.65	0	0	0	0	0	0	95.35	0	1.00
	C8	3.79	0	0	0	0	0	1.38	0.34	92.76	1.72
	C9	1.40	0	0	0	0	0	0.47	0.31	0.16	97.67

TABLE 3: Distraction classification results by training (AlexNet) and testing on skin segmented images.

Network	Accuracy [%]
No pixels' spatial info	93.14
Spatial info learned through active learning with 50% of data	93.53
Spatial info learned through active learning with 100% of data	93.62

TABLE 4: Distracted driver posture classification results on test data with unique drivers.

Network	Loss (NLL)	Accuracy [%]
VGG-16	1.246614	76.131
Resnet50	0.661483	81.695
InceptionV3	0.640018	90.068

unsatisfactory and needs better generalization against the varying lighting conditions happening during driving. Given that, in addition to several misdetections noted from the face (as in Figure 6) and hand detectors, we conclude that further work is needed to improve the accuracy of detection of these visual elements (skin, face, and hands). At the moment they

just add to the system complexity without improving the overall distraction identification accuracy.

The train/test data split presented in Table 1 causes a high correlation between training and testing data. Thus, it makes the distraction detection a much easier problem. This split is also followed in [54]. To present a more realistic setup, we adopt a more challenging strategy. We separate the data based on driving sessions rather than at random. In other words, test data split contains different drivers, cars, times of day, lighting conditions, driving conditions, and so on. Out of 44 drivers in our dataset, the training contains 38 drivers (12,555 samples), and the test data contains 6 drivers (1,923 samples).

The results of our genetically weighted ensemble of classifiers on the new split-by-driver test data is shown in Table 4. The confusion matrix is shown in Table 5. InceptionV3 achieves the best distraction identification accuracy of 90%. Figure 7 presents saliency maps generated for sample test images. The gradient of the winning distraction class is calculated and applied to input image. The gradient highlights the salient regions, i.e., the ones causing the most change and contributing to the network's decision. The drawn salient regions clearly show that our network is making its decision based on relevant features in the input camera images (i.e., hands, faces, neck, radio, driving wheel, and so on).

In a static image, a driver would appear in a "safe driving" posture, while contextually he/she might be distracted by doing some other activity. Making the distraction posture

TABLE 5: Confusion matrix of InceptionV3 network on test data with unique drivers.

		Predicted									
		C0	C1	C2	C3	C4	C5	C6	C7	C8	C9
Actual	C0	87.57	10.98	0	0.29	0	0	0.29	0.86	0	0
	C1	0	96.71	0	1.88	0	0	0	1.41	0	0
	C2	0	0	97.42	0	0	0	0	0.52	0.52	1.55
	C3	8.33	0	0	91.67	0	0	0	0	0	0
	C4	11.76	0	0	0	87.06	0	0	0	1.18	0
	C5	0	0	0	0	0	100	0	0	0	0
	C6	0.67	0	0	0	0	0	81.82	10.49	6.99	0
	C7	0	0	0	0	0	0	0	100	0	0
	C8	2.74	0	0.68	0	3.42	0	0	7.53	84.93	0.68
	C9	1.38	2.75	0	0	0	0	0	19.27	0	76.6



FIGURE 7: Saliency map demonstrates that our network makes its decision based on relevant features/regions in the input camera images.

identification decision based only on the current camera input image restricts the system from coping with such a problem. In this experiment, to classify a camera frame F_t at timestamp t , the network gives a decision C_t . After that, a final smoothed decision is deduced as the average of all

the preceding network decisions over the M seconds before t such that $C_{smoothed_t} = \sum_{i=t-M}^t C_i$.

In Figure 8, we evaluate the system accuracy at different values of M . $M = 1$ means that distraction posture identification decision is based only on the current camera input

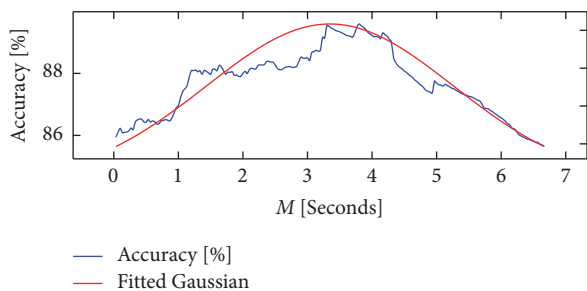


FIGURE 8: System accuracy evaluated at different values of history intervals M . At $M = 1$, current camera input image is solely used for distraction classification; no context is used. For $M > 1$, we average predictions of M frames to obtain distraction in the current frame. Up to a certain threshold, the more context incorporated in the prediction, the more accurate and robust the system. After that threshold, contextual information in the past is no longer relevant. Best accuracy is achieved at $M = 3.35$ seconds, which is the average of the fitted Gaussian curve.

image. A bell-curve-looking graph is obtained. It indicates that accuracy increases when more images from the past are considered. However, this happens until a threshold after which the past becomes irrelevant, thus, decreasing accuracy. The optimal value for M is found to be 3.35 seconds, which is the mean of a fitted Gaussian distribution to the results.

5.1. Analysis. In Table 2, we notice that the most confusing posture is the “safe driving”. This is due to the lack of temporal context in static images. In a static image, a driver would appear in a “safe driving” posture. However, contextually, he/she was distracted by doing some other activity. “Text Left” is mostly confused for “Talk Left” and vice versa. The same applies to “Text Right” and “Talk Right”. “Adjust Radio” is mainly confused for a “safe driving” posture. That is due to lack of the previously mentioned temporal context. Apart from safe driving, “Hair & Makeup” is confused for talking to passenger. That is because, in most cases, when drivers did their hair/makeup on the left side of their face, they needed to tilt their face slightly right (while looking at the frontal mirror). Thus, the network thought the person was talking to passenger. “Reach Behind” was confused for both talking to passenger and drinking. That makes sense as people tend to naturally look towards the camera while reaching behind. As for the drinking confusion, it is due to right-arm movement from the steering wheel to the back seat. A still image in the middle of that move could be easily mistaken for a drinking posture. “Drink” and “Talk to Passenger” postures were not easily confused with other postures as 98% and 97.67% of their images were correctly classified.

5.2. Real-Time System. The number of parameters in the deep models we adopted and benchmarked in 1 is huge for a real-time system. Our version of VGG16 [46], AlexNet [15], ResNet50 [45], and InceptionV3 [44] has 134.3M, 58.3M, 23.6M, and 21.8M parameters, respectively. Reference [54] adopts a real-time model that reduces the number of parameters in VGG16 to 15M, which is a marked improvement,

but is still a large number for a real-time system (in a self-driving car, for example). Hence we propose a version of NasNet Mobile [55] that reduces the number of parameters further to only 4.3M. An ensemble of two NasNet Mobile models [55] (original and skin-segmented networks) produce a satisfactory classification accuracy of 84.64%. Meanwhile, it still maintains a real-time performance on a CPU-based system.

For NasNet Mobile models fusion, we use a Multilayer Perceptron (MLP) to learn a class-based weighted ensemble from data. The advantage of our MLP-based fusion over the weighted voting system is that it does not assume that some classifier is better in absolute terms. An MLP learns more sophisticated dependencies; one classifier can be more accurate in discriminating a set of distraction classes, while another can be more accurate on another set of distraction classes. This also can happen under specific status of the ensemble resultant softmax class probability distributions. In our adopted MLP-based fusion, the vectors $C_i \in \mathbf{R}^{10}$, where $i = 1, \dots, N$, are concatenated and passed to an MLP. Note that, each C_i is a vector having 10 probabilities (corresponding to 10 distraction classes). Our MLP loss function evaluates the Negative Log Likelihood (NLL) loss over the training data in addition to a regularization term to prevent overfitting.

6. Conclusion

Distracted driving is a major problem leading to a striking number of accidents worldwide. Its detection is an important system component in semiautonomous cars. In this paper, we presented a robust vision-based system that recognizes distracted driving postures. We collected a novel publicly available distracted driver dataset that we used to develop and test our system. Our best model utilizes a genetically weighted ensemble of convolutional neural networks to achieve a 90% classification accuracy. We aim to provide a baseline performance for future research to benchmark against. We also showed that a simpler model (only using AlexNet) could operate in real-time and still maintain a satisfactory classification accuracy. Face, hands, and skin detection proved to improve classification accuracy in our ensemble. However, in a real-time setting, their performance overhead is much higher than their contribution.

In a future work, we need to devise a better face, hands, and skin detector. We would need to manually label hand and face proposals and use them to train a Fast-RCNN (or, any other object detector) to localize both faces and hands in one shot and evaluate it against our existing CNN-based localization method.

Data Availability

The data used to support the findings of this study are available from the corresponding authors upon request. Our Distracted Driver dataset is publicly available at the project website link: https://heshameraqi.github.io/distraction_detection.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the German Research Foundation (DFG) and the Technical University of Munich within the Open Access Publishing Funding Programme.

Supplementary Materials

A video showing our proposed method accuracy in run-time. The probabilities for the top 3 predicted classes are displayed along with the system input camera video. (*Supplementary Materials*)

References

- [1] Y. Liang, *Detecting Driver Distraction [Theses and Dissertations]*, 2009.
- [2] A. Downs, *Why Traffic Congestion Is Here to Stay... And Will Get Worse*, ACCESS Magazine, 2004.
- [3] W. H. Organization, "World health statistics 2017: monitoring health for the sdgs, sustainable development goals," 2017.
- [4] G. W. H. Organization, "Global health estimates 2015: Deaths by cause, age, sex, by country and by region, 2000-2015," 2016, http://www.who.int/healthinfo/global_burden_disease/estimates/en/index1.html.
- [5] D. Strickland, "How autonomous vehicles will shape the future of surface transportation," *Testimony of the Honorable David L. Strickland Administrator National Highway Traffic Safety Administration House Committee on Transportation and Infrastructure Subcommittee on Highways and Transit*, 2013.
- [6] J. D. Lee, "Driving safety," *Reviews of Human Factors and Ergonomics*, vol. 1, no. 1, pp. 172–218, 2016.
- [7] T. M. Pickrell, H. R. Li, and S. KC, "Traffic safety facts," 2016, <https://www.nhtsa.gov/risky-driving/distracted-driving>.
- [8] U. D. o. H. & H. Services, "Distracted Driving," 2016, <https://www.cdc.gov/motorvehiclesafety/distracted.driving/>.
- [9] A. Eriksson and N. A. Stanton, "Takeover time in highly automated vehicles: Noncritical transitions to and from manual control," *Human Factors: The Journal of the Human Factors and Ergonomics Society*, vol. 59, no. 4, pp. 689–705, 2017.
- [10] H. M. Eraqi, M. N. Moustafa, and J. Honer, "End-to-end deep learning for steering autonomous vehicles considering temporal dependencies," in *Proceeding of the Machine Learning for Intelligent Transportation Systems Workshop in the 31st Conference on Neural Information Processing Systems (NIPS)*, Montreal, Canada, December 2017.
- [11] H. M. Eraqi, J. Honer, and S. Zuther, "Static free space detection with laser scanner using occupancy grid maps authors," in *Proceeding of the 20th International Conference on Intelligent Transportation Systems (ITSC)*, IEEE, Yokohama, Japan, October 2017.
- [12] H. M. Eraqi, Y. Emad Eldin, and M. N. Moustafa, "Reactive collision avoidance using evolutionary neural networks," in *Proceedings of the 8th International Conference on Evolutionary Computation Theory and Applications*, vol. 1, pp. 251–257, SciTePress, Porto, Portugal, November 2016.
- [13] A. Fernández, R. Usamentiaga, J. L. Carús, and R. Casado, "Driver distraction using visual-based sensors and algorithms," *Sensors*, vol. 16, no. 11, article no 1805, 2016.
- [14] S. S. Farfade, M. J. Saberian, and L. Li, "Multi-view face detection using deep convolutional neural networks," Cornell University Library, 2015, <https://arxiv.org/abs/1502.02766>.
- [15] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proceedings of the 26th Annual Conference on Neural Information Processing Systems (NIPS '12)*, pp. 1097–1105, Lake Tahoe, Nev, USA, December 2012.
- [16] A. Dosovitskiy, P. Fischer, J. T. Springenberg, M. Riedmiller, and T. Brox, "Discriminative unsupervised feature learning with exemplar convolutional neural networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 9, pp. 1734–1747, 2016.
- [17] O. Russakovsky, J. Deng, H. Su et al., "Imagenet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.
- [18] D. Erhan, C. Szegedy, A. Toshev, and D. Anguelov, "Scalable object detection using deep neural networks," in *Proceedings of the 27th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2014*, pp. 2155–2162, USA, June 2014.
- [19] A. Oliva and A. Torralba, "Modeling the shape of the scene: a holistic representation of the spatial envelope," *International Journal of Computer Vision*, vol. 42, no. 3, pp. 145–175, 2001.
- [20] Y. Abouelnaga, "Action annotation tool," 2017, <https://github.com/devyhia/action-annotation>.
- [21] R. A. Berri, A. G. Silva, R. S. Parpinelli, E. Girardi, and R. Arthur, "A pattern recognition system for detecting use of mobile phones while driving," in *Proceedings of the 9th International Conference on Computer Vision Theory and Applications, VISAPP 2014*, vol. 2, pp. 411–418, IEEE, Portugal, January 2014.
- [22] Y. Artan, O. Bulan, R. P. Loce, and P. Paul, "Driver cell phone usage detection from HOV/HOT NIR images," in *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops, (CVPRW '14)*, pp. 225–230, USA, June 2014.
- [23] C. Craye and F. Karray, "Driver distraction detection and recognition using RGB-D sensor," 2015, <https://arxiv.org/abs/1502.00250>.
- [24] X. Zhang, N. Zheng, F. Wang, and Y. He, "Visual recognition of driver hand-held cell phone use based on hidden CRF," in *Proceedings of the 2011 IEEE International Conference on Vehicular Electronics and Safety, ICVES 2011*, pp. 248–251, China, July 2011.
- [25] T. H. Le, Y. Zheng, C. Zhu, K. Luu, and M. Savvides, "Multiple scale faster-rcnn approach to driver's cell-phone usage and hands on steering wheel detection," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 46–53, IEEE, Las Vegas, Nev, USA, June 2016.
- [26] N. Das, E. Ohn-Bar, and M. M. Trivedi, "On performance evaluation of driver hand detection algorithms: challenges, dataset, and metrics," in *Proceedings of the 18th IEEE International Conference on Intelligent Transportation Systems (ITSC '15)*, pp. 2953–2958, Spain, September 2015.
- [27] K. Seshadri, F. Juefei-Xu, D. K. Pal, M. Savvides, and C. P. Thor, "Driver cell phone usage detection on strategic highway research program (SHRP2) face view videos," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, (CVPRW '15)*, pp. 35–43, USA, June 2015.

- [28] S. Martin, E. Ohn-Bar, A. Tawari, and M. M. Trivedi, "Understanding head and hand activities and coordination in naturalistic driving videos," in *Proceedings of the 25th IEEE Intelligent Vehicles Symposium, IV 2014*, pp. 884–889, IEEE, USA, June 2014.
- [29] E. Ohn-Bar, S. Martin, and M. M. Trivedi, "Driver hand activity analysis in naturalistic driving studies: challenges, algorithms, and experimental studies," *Journal of Electronic Imaging*, vol. 22, no. 4, Article ID 041119, 2013.
- [30] E. Ohn-Bar, S. Martin, A. Tawari, and M. M. Trivedi, "Head, eye, and hand patterns for driver activity recognition," in *Proceedings of the 22nd International Conference on Pattern Recognition (ICPR)*, pp. 660–665, Stockholm, Sweden, August 2014.
- [31] E. Ohn-Bar and M. Trivedi, "In-vehicle hand activity recognition using integration of regions," in *Proceedings of the 2013 IEEE Intelligent Vehicles Symposium, IEEE IV 2013*, pp. 1034–1039, Australia, June 2013.
- [32] C. H. Zhao, B. L. Zhang, J. He, and J. Lian, "Recognition of driving postures by contourlet transform and random forests," *IET Intelligent Transport Systems*, vol. 6, no. 2, pp. 161–168, 2012.
- [33] C. Zhao, Y. Gao, J. He, and J. Lian, "Recognition of driving postures by multiwavelet transform and multilayer perceptron classifier," *Engineering Applications of Artificial Intelligence*, vol. 25, no. 8, pp. 1677–1686, 2012.
- [34] C. Zhao, B. Zhang, J. Lian, J. He, T. Lin, and X. Zhang, "Classification of driving postures by support vector machines," in *Proceedings of the Sixth International Conference on Image and Graphics (ICIG '11)*, pp. 926–930, Hefei, China, August 2011.
- [35] C. H. Zhao, B. L. Zhang, X. Z. Zhang, S. Q. Zhao, and H. X. Li, "Recognition of driving postures by combined features and random subspace ensemble of multilayer perceptron classifiers," *Neural Computing and Applications*, vol. 22, no. 1, pp. 175–184, 2013.
- [36] C. Yan, F. Coenen, and B. Zhang, "Driving posture recognition by joint application of motion history image and pyramid histogram of oriented gradients," *International Journal of Vehicular Technology*, vol. 2014, Article ID 719413, 11 pages, 2014.
- [37] F. Coenen, B. Zhang, and C. Yan, "Driving posture recognition by convolutional neural networks," *IET Computer Vision*, vol. 10, no. 2, pp. 103–114, 2016.
- [38] I. Sultan, "Academic purposes?" 2016, <https://www.kaggle.com/c/state-farm-distracted-driver-detection/discussion/20043#117982>.
- [39] S. Yan, Y. Teng, J. S. Smith, and B. Zhang, "Driver behavior recognition based on deep convolutional neural networks," in *Proceedings of the 12th International Conference on Natural Computation and 13th Fuzzy Systems and Knowledge Discovery (ICNC-FSKD)*, pp. 636–641, Changsha, China, August 2016.
- [40] T. A. U. in Cairo (AUC), "Auc distracted driver dataset," 2019, https://heshameraqi.github.io/distraction_detection.
- [41] Y. Abouelnaga, H. M. Eraqi, and M. N. Moustafa, "Real-time distracted driver posture classification," in *Proceedings of the Machine Learning for Intelligent Transportation Systems Workshop in the 32nd Conference on Neural Information Processing Systems (NeuroIPS)*, Long Beach, Calif, USA, December 2018.
- [42] ASUS, "Zenfone selfie smartphone (model zd551kl)," 2015, https://www.asus.com/eg-en/Phone/ZenFone_Selfie-ZD551KL/.
- [43] Sony, "Ds325 camera datasheet," 2015, https://www.sony-depthsensing.com/Portals/0/Download/WEB_20120907_SK_DS325_Datasheet_V2.1.pdf.
- [44] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016*, pp. 2818–2826, July 2016.
- [45] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016*, pp. 770–778, July 2016.
- [46] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, <https://arxiv.org/abs/1409.1556>.
- [47] S. Phung, A. Bouzerdoum, and D. Chai, "Skin segmentation using color and edge information," in *Proceedings of the Seventh International Symposium on Signal Processing and Its Applications, 2003. Proceedings.*, pp. 525–528 vol.1, Paris, France, July 2003.
- [48] R. Bhatt and A. Dhall, "Skin segmentation dataset," *UCI Machine Learning Repository*, 2010.
- [49] H. Li, Z. Lin, X. Shen, J. Brandt, and G. Hua, "A convolutional neural network approach for face identification," in *Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5325–5334, Boston, Mass, USA, June 2015, http://users.eecs.northwestern.edu/~xsh835/assets/cvpr2015_cascnn.pdf.
- [50] M. Kostinger, P. Wohlhart, P. M. Roth, and H. Bischof, "Annotated facial landmarks in the wild: A large-scale, real-world database for facial landmark localization," in *Proceedings of the IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, pp. 2144–2151, IEEE, Barcelona, Spain, November 2011.
- [51] S. Bambach, S. Lee, D. J. Crandall, and C. Yu, "Lending a hand: detecting hands and recognizing activities in complex egocentric interactions," in *Proceedings of the 15th IEEE International Conference on Computer Vision, (ICCV '15)*, pp. 1949–1957, Chile, December 2015.
- [52] Tensorflow, "TensorFlow-Slim image classification library," 2018, <https://github.com/tensorflow/models/tree/master/slim>.
- [53] L. Rokach, "Ensemble-based classifiers," *Artificial Intelligence Review*, vol. 33, no. 1-2, pp. 1–39, 2010.
- [54] B. Baheti, S. Gajre, and S. Talbar, "Detection of distracted driver using convolutional neural network," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 1032–1038, IEEE, Salt Lake City, UT, USA, June 2018.
- [55] B. Zoph, V. Vasudevan, J. Shlens, and Q. V. Le, "Learning transferable architectures for scalable image recognition," *Computer Vision and Pattern Recognition*, vol. 2, no. 6, 2017, <https://arxiv.org/abs/1707.07012>.



Hindawi

Submit your manuscripts at
www.hindawi.com

