A Deep Learning Based Framework for Distracted Driver Detection

Swadesh Kumar Maurya and Ayesha Choudhary(✉)

School of Computer and Systems Sciences,
Jawaharlal Nehru University, New Delhi, India
swades89_scs@jnu.ac.in, ayeshac@mail.jnu.ac.in

Abstract. In this paper, we propose a novel, real-time, deep learning-based framework for distracted driver detection for driver Advanced Driver Assistance Systems (ADAS). We assume that the camera is assumed to be mounted inside the vehicle such that the side view of the driver is in view. Distracted driving is a serious problem leading to a large number of serious and even fatal road accidents worldwide every year. We propose a deep learning architecture that takes as input the captured images of the driver and classifies and recognizes the various distracted driving behaviors. It also recognizes if the driver is not distracted and is alert. The experiments are performed on the publicly available State Farm Distracted Driver Detection (SFDDD) dataset [1] which has 9 classes of distracted driver behavior and one class of alert driving. The training time for the proposed framework is minimal and approach works in real-time. Our experimental results show that our proposed framework is robust and performs better than the state-of-the-art approaches on this dataset.

Keywords: Driver assistance · Distracted driver detection · Deep learning · Driver behavior · Smart vehicle · Road safety

1 Introduction

In this paper, we propose a novel, real-time framework for detecting distracted driving by placing a camera inside the vehicle such that the side view of the driver is visible. In recent years, there has been immense progress in the area of Intelligent Transportation System (ITS) that focuses on developing an Advanced Driver Assistance System (ADAS) to provide a safe driving environment. In this area, the problem of unexpected behavior and distracted driver on the road is very important since it may lead to serious and fatal accidents. According to the CDC motor vehicle safety division [2], one in five car accidents is caused by a distracted driver. Sadly, this translates to 425,000 people injured and 3,000 people killed by distracted driving every year across the US only.

There are three main types of driver distractions: (a) Visual: taking eyes off the road by the driver; (b) Manual: taking hands off the wheel that distracts

R. V. Babu et al. (Eds.): NCVPRIPG 2019, CCIS 1249, pp. 163–173, 2020.
https://doi.org/10.1007/978-981-15-8697-2_15

Fig. 1. Distracted driver dataset sample visualization for all 10 classes (a–j), the classes detail of subfigure are (a) c0: Safe driving, (b) c1: Texting-right, (c) c2: Talking on the phone-right, (d) c3: Texting-left, (e) c4: Talking on the phone-left, (f) c5: Operating the radio, (g) c6: Drinking (h) c7: Reaching behind, (i) c8: Hair and makeup, (j) c9: Talking to passenger.

a driver's mind from driving; (c) Cognitive: taking the mind off from driving, leading to unattentive driving. Distracted driver detection is a major challenge to perform in an ADAS for improving driving conditions. In this work, we focus on visual and manual distractions and propose a Driver Assistance System that tracks the behavior of the driver while he is driving and alerts the driver if he/she does an unexpected task thereby increasing the chance of accidents. In autonomous driving systems, the driver needs to be well-prepared to take over the controls, whenever required. In such cases also, the distracted driver detection is an important issue and the driver should be alerted if he/she is distracted and not well-prepared to take control immediately in case the need arises.

There are various challenges in distracted driver detection such as illumination variation, occlusion, camera perspective may vary, day and night driving environments as well as the different clothing and the driver dependent physical characteristics. Based on the kind of distraction such as visual, manual or cognitive, different challenges arises. In our work, we focus on the visual and manual types of distractions of the driver.

We propose a deep learning-based novel architecture that builds upon a pre-trained deep learning model, by further adding new layers to improve the performance. The deep learning models about the very large ImageNet dataset [7] are used to perform the training on the driver distraction dataset. We evaluate our approach and use different pre-trained models to compare and analyze the performance and convergence rate of individual models. This allows us to improve and find the optimal performance that requires the least amount of training and computation time. We use the State Farm Distracted Driver Detection Dataset [1], which has images of different drivers performing 9 classes of distracted behavior and 1 class of alert driving behavior (samples have shown in Fig. 1). We compare our proposed approach with the state of the art methods that exist on this dataset. The paper is structured as follows: In Sect. 2, the related work is discussed. We discuss our proposed work in Sect. 3 and the experimental results in Sect. 4. Finally, we conclude in Sect. 5.

2 Related Work

Detection of distracted driving is getting attention in the research community and the industry. Various vision and sensor-based approaches are proposed in this area. In this work, we focused on the techniques that use vision-based approaches. The detection of distracted driver behavior (i.e. driver drowsiness, lane departure, talking on phone, looking back, etc.) using a camera mounted inside the vehicle is an active area of research as Advanced Driver Assistance System (ADAS), various computer vision-based and machine learning-based methods are applied to extract features such as eye-tracking, driver posture, cellphone usage, etc. in driving images or video. The approaches proposed in [3,4] are focused on extracting features such as eye-tracking, driver posture, cellphone usage, etc. in driving image scene or video. These approaches are used in the distracted driving behavior analysis in the state of art.

The Convolutional Neural Networks (CNN) and deep learning approaches such as [5,6] are used in various work to perform the driver behavior analysis. In [8] the VGG-16 architecture is modified and used for the classification such that the system not only detects distracted driving behavior but also finds the type of distraction in the scene. Research in distracted driving behavior has also focused on the face and hand position of the driver in a naturalistic driving environment [9,11,12]. They focus on the detecting hand region to identifying the type of activities such as adjusting radio, mirror, operating gear.

Similarly, in Le et al. [10], identification of distracted driving is based on the position of the hand and cellphone usage. This uses multi-scale faster-RCNN for detecting objects such as cellphones, hands, steering wheel, etc. and classifies the behavior based on the position of objects. In [17], deep learning approach is used to classify the driver behaviour on the SFDDD dataset [1]. They have used AlexNet with Softmax and Triplet Loss to perform the classification task and achieved the 98.7% accuracy. Zhao et al. [13] proposed a distracted driver dataset with the side view of the driver having only four activities: safe driving, operating shift lever, eating and talking on the cellphone. In this paper, we propose a novel, real-time framework for the distracted driver detection and develop a new architecture that improves upon the feature extracted from a pre-trained network. We describe our proposed in the next proposed work section.

3 Proposed Work

In this work, we propose a novel architecture for classification using deep learning. The main challenge in classification is to be able to extract important features for better classification performance. For this, we use a pre-trained deep learning model as the base layer. In our implementation, we used the models and their pre-trained weights on ImageNet dataset [7] so that the optimal features can be learned from the images faster and with minimal training. Our proposed full model architecture is shown in Fig. 2 with all the layers, dropout, batch normalization, and other activation function structure.

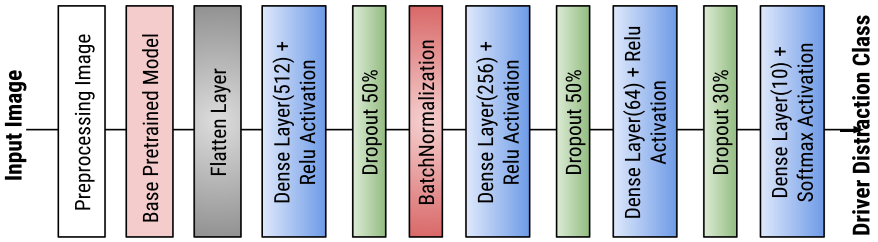

Fig. 2. Our proposed model architecture for distracted driver detection, where the Base Pre-trained Model layer can be replaced with any of the pre-trained model by Resnet50 [14], Inceptionv3 [14], InceptionResnetv2 [16] and Mobilenet [15].

As shown in Fig. 2, we define the following layers in our model. We consider the pre-trained network as the base layer, then add the 'Flatten layer' to ensure that all the features are converted into a single vector. The next layers are the 'Dense' layer with ReLu activation function followed by the 'Dropout' layer. We use dropout to ensure that generalized features are learned to overcome the problem of overfitting in the model. These 'Dense' layers and 'Dropout' are repeated with a reduced number of dense nodes as the number of features is also decreased from the top layers. We also perform Batch normalization after the first dense layer to ensure normalization of the features obtained from the initial layers.

The last output layer is meant to classify among K categories with a SoftMax activation function given by Eq. 3, that assigns conditional probabilities (given x) of each categories. We use the stochastic gradient descent (SGD) optimizer to train the model networks because new optimizers such as Adam and Nadam are not able to learn optimal weight at the time of training as compared to the SGD optimizer. The cross-entropy loss $\mathcal{L}$ used in the model to optimize the model weight and maximize the accuracy of the classification is given by Eq. 1.

$$\mathcal{L}\left(y_i, \hat{y}_i\right) = -\sum_{i=1}^{K} y_i \log\left(\hat{y}_i\right) \tag{1}$$

The each input batch sample is preprocessed as given in Eq. 2, initially it is mean centered on the average of the input batch and then normalized.

$$z = \frac{x-\mu}{\sigma}, \mu = \frac{1}{N}\sum_{j=1}^{N}(x_j),\ \sigma = \sqrt{\frac{1}{N}\sum_{j=1}^{N}(x_j-\mu)^2} \tag{2}$$

$$S_i = \frac{\exp(\mathbf{w}_i^\top \mathbf{x})}{\sum_{k=1}^{K}\exp(\mathbf{w}_k^\top \mathbf{x})} = \frac{\exp(Z_i)}{\sum_{k=1}^{K}\exp(Z_k)} \tag{3}$$

S_i defines the Softmax activation function on i^{th} class, w are the neuron connection's weight and x is the feature at dense layers.

We applied the early stopping criteria of 15 epoch patience to training whether it gets optimized or not. The technique 'Reduce on Plateau' is applied to get the advantage of reducing the learning rate by a factor of 0.95 once learning stagnates. This callback monitors a loss value and if no improvement is seen for patience or a specified number of epochs, the learning rate is reduced for future epochs. The pre-trained base model which we select as the base layer feature extraction, are ResNet50 [14],InceptionV3 [5], InceptionResnetv2 [16], and Mobilenet [15]. Each model is modified with the given architecture defined in Fig. 2 to improve classification and optimize the performance. The training results and all performance matrices are shown in Sect. 4 for each proposed model. If we consider the input image directly as input in the architecture, it may lead to an increase in computations as well as training time to learn the features, therefore, the input images are re-sampled and re-sized to fit in the base model architecture and meet the hardware requirements of the system. The image samples are then normalized by feature-wise mean centering, which sets the input means to 0 over the dataset feature-wise as shown in Eq. 2. The x here represents the complete set of images in a batch and μ is the mean of the complete data in the batch, and z is the final feature that is normalized and centered around the mean. The data augmentation significantly increases the diversity of data available and captures data invariance for training models. The data augmentation is performed with a rotation range of 30° and horizontal flip of the image so the scene can add diversity and variations in the dataset and consider the various cases of camera angle changes.

4 Experimental Results and Discussion

In our experiments, we use the State Farm Distracted Driver Detection(SFDDD) dataset [1]. It is publicly available with images taken in a car where the driver is acting on the set of activities such as texting, eating, talking on the phone, makeup, reaching behind, etc. The images are taken from a camera inside the vehicle such that the side view of the driver is visible. The dataset consists of 10 classes (Safe driving + 9 distracting behaviors) to predict and their performing actions are: c0: Safe driving; c1: Texting - right; c2: Talking on the phone - right; c3: Texting - left; c4: Talking on the phone - left; c5: Operating the radio; c6: Drinking; c7: Reaching behind; c8: Hair and makeup; c9: Talking to passenger(s). In our experiments, we perform the training with the large set of driver distraction dataset (Fig. 1), for which 22,000 labeled images are available. We have used 16,000 images in our experiments, out of which 10,000 images are

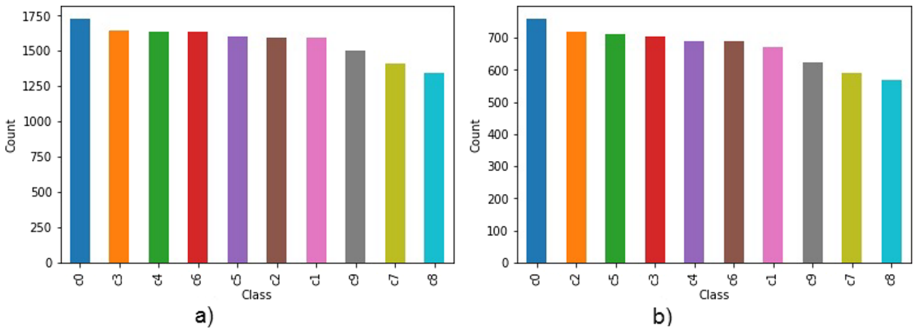

Fig. 3. Class distribution in the (a) training split data and (b) testing split data class distribution.

taken and split into training and validation set in the ratio of 70:30 such that 7000 images are taken for training and the rest 3000 images are for validation. The remaining 6000 images are reserved for testing to evaluate model performance. The reason to take the subset of 16000 images so that training time can be reduced and performance can be analyzed with this limited training set and evaluated on the large test set, with these limitations also the system performed accurately at testing phase, it shows that this amount of data is enough to learn this distracted driver classification task. The detailed class-wise training and testing data distribution are shown in Fig. 3.

The results are shown after the training and evaluation of the optimized model on the test data of 6000 images. The model with the best-achieved accuracy and loss are compared and the test data evaluation matrices are also shown with class-wise performance and number of support samples. The training and validation results of the different models as a comparison between the training and the validation accuracy and loss in performance with their corresponding epoch are shown in Fig. 5. Our experiments discover that even with large training datasets the proposed model with Resnet50 and Mobilenet models as base layers converge with only 49 and 46 epochs, respectively, with a very low number of trainable parameters. This is in comparison to our proposed model with Inceptionv3 and InceptionResnetv2 as base layers, in term of number of epoch to train and fit the model on the given training dataset, and also in terms of the number of trainable parameters in the model, as can be seen by the parameter analysis shown in Table 1.

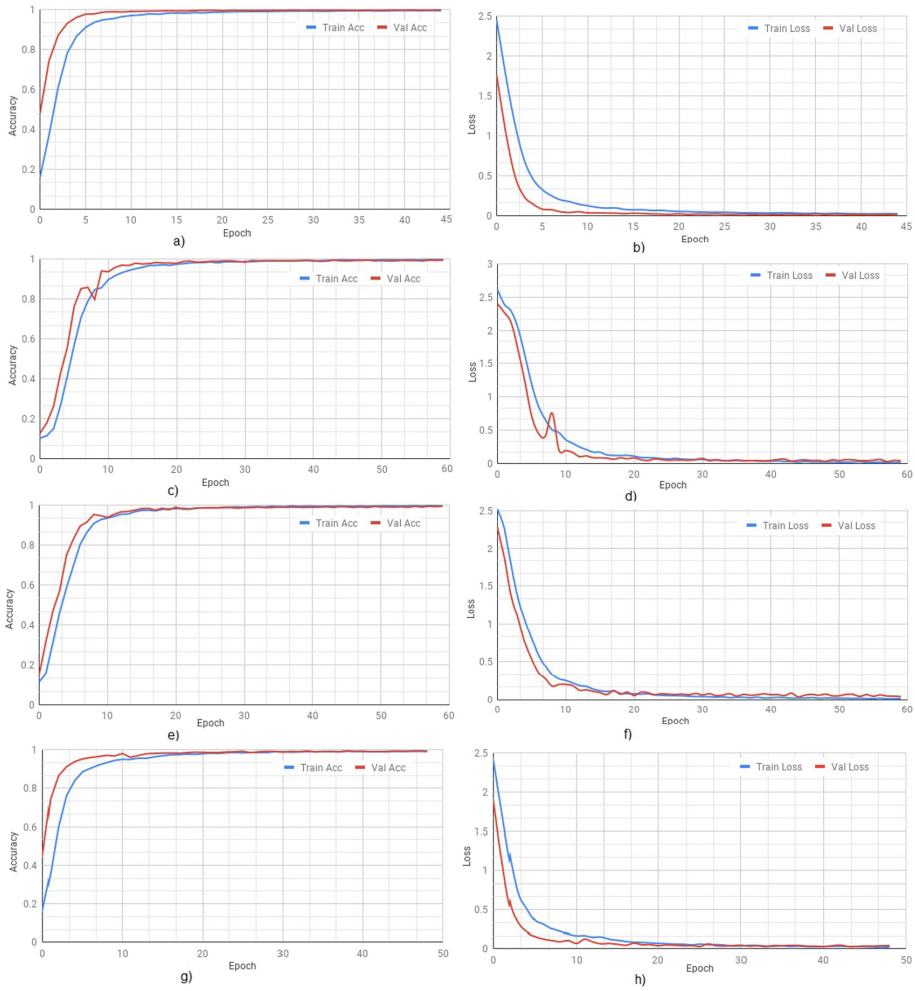

Fig. 4. Comparison of the training and validation accuracy and loss on each epoch, is shown in the figure, subfigure a) and b) shows the Resnet50 accuracy and loss plot, subfigure c) and d) shows the Inceptionv3 accuracy and loss plot, subfigure e) and f) shows the InceptionResnetv2 accuracy and loss plot, subfigure g) and h) shows the Mobilenet accuracy and loss plot.

Table 1. Model comparison and performance parameters

Proposed model using	#Parameter	Input Size	#Epoch	Accuracy	Loss	Test time(6000)
ResNet50	24,733,130	200×200	46	99.75	0.0094	104 s, 17 ms/sample
Inceptionv3	38,730,986	200×200	60	99.39	0.0413	73 s, 12 ms/sample
InceptionResnetv2	67,009,066	200×200	59	99.19	0.0491	75 s, 12 ms/sample
MobileNet	22,231,306	192×192	49	99.23	0.0410	34 s, 6 ms/sample

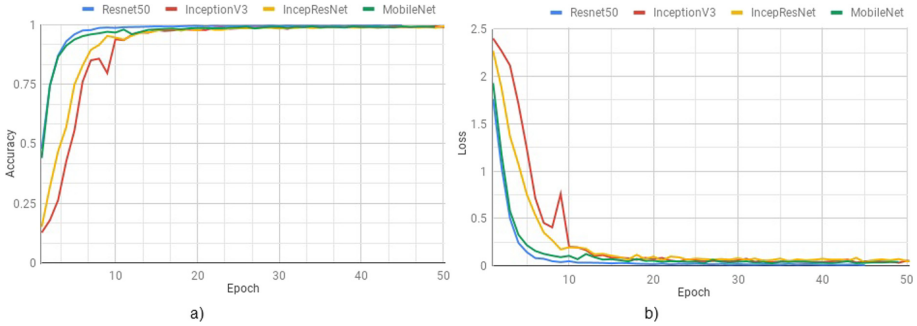

Fig. 5. Training results comparison in subfig a) and b) on all four proposed models. a) Epoch vs Accuracy comparison plot on proposed models on Validation set. b) Epoch vs Loss comparison plot on proposed models on Validation set.

Table 2. Class wise f1-Scores on the different proposed models with the number of support

Class	Resnet50	Inceptionv3	InceptionResnetv2	MobileNet	Support
c0	0.99	0.98	0.98	0.98	682
c1	0.99	0.99	0.99	1.00	602
c2	0.99	0.99	0.99	0.99	651
c3	0.99	1.00	1.00	0.99	626
c4	0.99	0.98	0.99	0.99	612
c5	0.98	0.99	0.99	0.99	628
c6	0.99	0.99	0.99	0.98	617
c7	0.98	1.00	1.00	1.00	522
c8	0.99	0.99	0.99	0.98	506
c9	1.00	0.98	0.98	0.98	554
micro-avg	0.99	0.99	0.99	0.99	6000
macro-avg	0.99	0.99	0.99	0.99	6000
weighted-avg	0.99	0.99	0.99	0.99	6000

Table 3. Summary of distracted driver detection results and comparison on State Farm Distracted Driver Detection dataset [1]

Model	Source	Accuracy%
AlexNet+Softmax Loss[17]	Original	96.8
AlexNet+Triplet Loss[17]	Original	98.7
Original VGG[8]	Original	94.44
VGG with Regularization[8]	Original	96.31
Modified VGG[8]	Original	95.54
Our proposed model using Resnet50	Original	99.75
Our proposed model using Inceptionv3	Original	99.39
Our proposed model using InceptionResnetv2	Original	99.19
Our proposed model using Mobilenet	Original	99.23

The individual model training results in terms of accuracy and loss values is shown in Fig. 4. The accuracy and loss plot of our proposed model using Resnet50 as the base layer is shown in Subfig. 4(a)–(b), of using Inceptionv3 as the base layer is shown in Subfig. 4(c)–(d), of using InceptionResnetv2 as the base layer is shown in Subfig. 4(e)–(f), and using Mobilenet as the base layer is shown in Subfig. 4(g)–(h). In terms of time on the test dataset, we perform the pre-processed test-set batch of 6000 samples already loaded in the memory, and with GPU mode, as mentioned in the Table 1. The last column 'Test time(6000)' shows the evaluation time on the complete 6000 samples are 104 s, 73 s, 75 s, 34 s on our proposed architecture with Resnet50, Inceptionv3, InceptionResnetv2, mobile net as base layers, respectively. This indicates that the performance of our architecture with the Mobilenet model as the base layers is much faster and more accurate as compared to the other three models. The micro-average f1-score on all four models is mentioned in Table 2 where the f1-score is given for each class separately, so we can analyze that not only overall classification performance is good but the individual classwise performance is also better for each trained models. The Fig. 6, shows the confusion matrix with all 10 classes of driver distraction for each model separately. The principal diagonal of the confusion matrix with dense values shows the better performance of our proposed model. The summary of distracted driver detection results and comparison with earlier approaches and our proposed approaches is shown in Table 3 shows better performance than other proposed models.

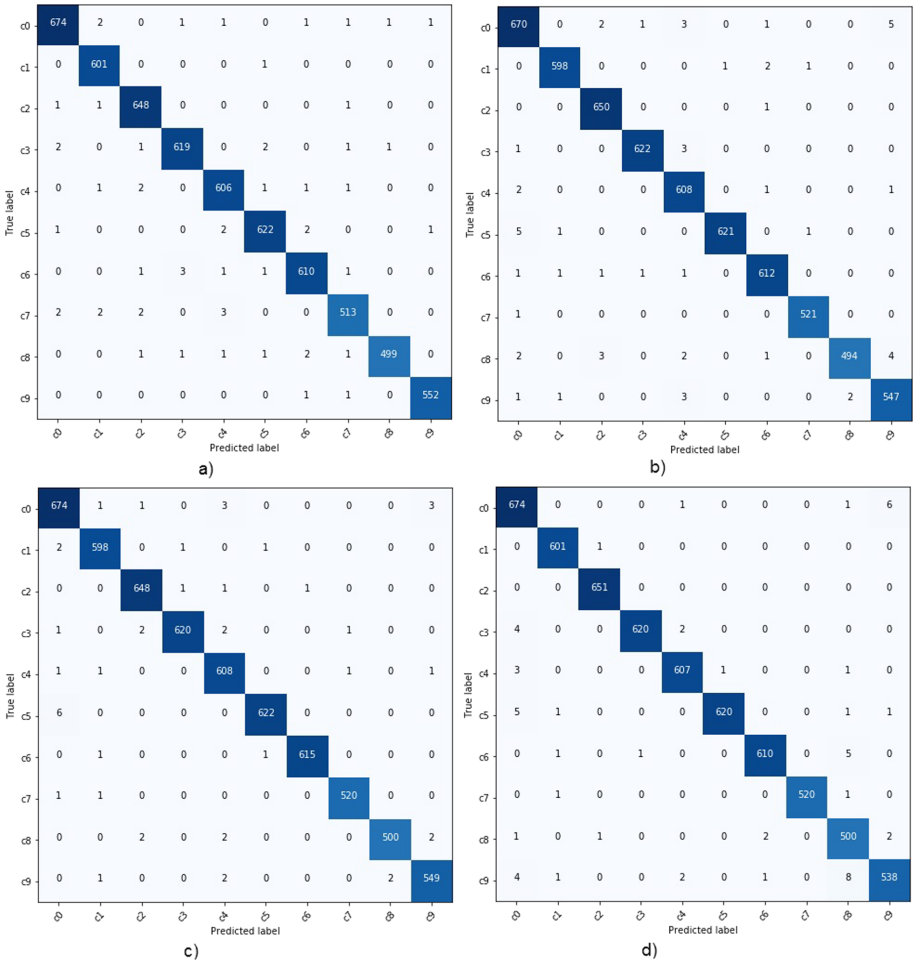

Fig. 6. Confusion matrix results on the test data by our four proposed models using different pretrained model as the base model. a) Resnet50 as base model, b) Inceptionv3 as base model, c) InceptionResnetv2 as base model, d) MobileNet as base model.

5 Conclusion

Driver distraction is a serious problem leading to a large number of road accidents worldwide. Hence, the detection of a distracted driver is important for the safety and security of the driver as well as the passengers. Our work focuses on the detection of the distracted driver from the scene captured from inside the vehicle. In this paper, we have proposed a deep learning-based classification model that uses the state of the art pre-trained deep learning models fine-tuned for distracted driver classification as the base layer. We use the publicly available State Farm Distracted Driver Detection dataset which has a large set of images

for 10 classes. The training and testing results show that our proposed model has high classification performance, that is, up to 99.75% in different models. Our proposed architecture is robust and fast and works well in real-time scenarios.

References

1. State farm distracted driver detection. https://www.kaggle.com/c/state-farm-distracted-driver-detection
2. CDC motor vehicle safety division. https://www.cdc.gov/motorvehiclesafety/distracted_driving
3. Streiffer, C., et al.: Darnet: a deep learning solution for distracted driving detection. In: Proceedings of the 18th ACM/IFIP/USENIX Middleware Conference: Industrial Track. ACM (2017)
4. You, C.-W., et al.: Carsafe app: alerting drowsy and distracted drivers using dual cameras on smartphones. In: Proceedings of the 11th Annual International Conference on Mobile Systems, Applications, and Services. ACM (2013)
5. Szegedy, C., et al.: Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2016)
6. Szegedy, C., et al.: Going deeper with convolutions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2015)
7. Deng, J., et al.: Imagenet: a large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition. IEEE (2009)
8. Baheti, B., Suhas, G., Sanjay, T.: Detection of distracted driver using convolutional neural network. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (2018)
9. Das, N., Ohn-Bar, E., Trivedi, M.M.: On performance evaluation of driver hand detection algorithms: challenges, dataset, and metrics. In: 2015 IEEE 18th International Conference on Intelligent Transportation Systems. IEEE (2015)
10. Hoang Ngan Le, T., et al.: Multiple scale faster-RCNN approach to driver's cell-phone usage and hands on steering wheel detection. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW). IEEE (2016)
11. Martin, S., et al.: Understanding head and hand activities and coordination in naturalistic driving videos. In: 2014 IEEE Intelligent Vehicles Symposium Proceedings. IEEE (2014)
12. Ohn-Bar, E., Trivedi, M.: In-vehicle hand activity recognition using integration of regions. In: 2013 IEEE Intelligent Vehicles Symposium (IV). IEEE (2013)
13. Zhao, C.H., et al.: Recognition of driving postures by contourlet transform and random forests. IET Intell. Transport Syst. **6**(2), 161–168 (2012)
14. He, K., et al.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2016)
15. Howard, A.G., et al.: Mobilenets: efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861 (2017)
16. Szegedy, C., et al.: Inception-v4, inception-resnet and the impact of residual connections on learning. In: Thirty-First AAAI Conference on Artificial Intelligence (2017)
17. Okon, O.D., Meng, L.: Detecting distracted driving with deep learning. In: Ronzhin, A., Rigoll, G., Meshcheryakov, R. (eds.) ICR 2017. LNCS (LNAI), vol. 10459, pp. 170–179. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-66471-2_19