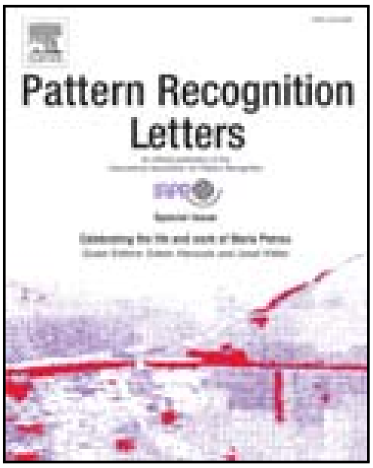
Accepted Manuscript



Detecting Distraction of drivers using Convolutional Neural Network

Sarfaraz Masood , Abhinav Rai , Aakash Aggarwal , M.N. Doja , Musheer Ahmad

PII: S0167-8655(17)30469-5
DOI: 10.1016/j.patrec.2017.12.023
Reference: PATREC 7039

To appear in: *Pattern Recognition Letters*

Received date: 4 December 2016
Revised date: 14 October 2017
Accepted date: 29 December 2017

Highlights

- A model to detect distraction of drivers.
- Utilizes convolutional neural networks to detect the activity being performed by the driver.
- The model is able to differentiate between the types of distractions as well**.**
- High detection accuracy of 99% achieved over a large dataset.
- The model proposed, takes significantly less training time and delivers high test accuracy.

Pattern Recognition Letters
journal homepage: www.elsevier.com

Detecting Distraction of drivers using Convolutional Neural Network

Sarfaraz Masood*, Abhinav Rai, Aakash Aggarwal, M. N. Doja and Musheer Ahmad

Department of Computer Engineering, Jamia Millia Islamia, New Delhi India

ABSTRACT

With the intervention of social media and internet technology, people are getting more and more careless and distracted while driving which is having a severe detrimental effect on the safety of the driver and his fellow passengers. To provide an effective solution, this paper put forwards a Machine Learning model using Convolutional Neural Networks to not only detect the distracted driver but also identify the cause of his distraction by analyzing the images obtained using the camera module installed inside the vehicle. Convolutional neural networks are known to learn spatial features from images, which can be further examined by fully connected neural networks. The experimental results show a 99% average accuracy in distraction recognition and hence strongly support that our Convolutional neural networks modelcan be used to identify distraction among the drivers.

* Tel.: +91-11-29860281; e-mail: smasood@jmi.ac.in

1. Introduction

The safety of people on road while driving has become a serious concern worldwide, over the past few years. As per the official US government website [22] on distracted driving, in 2015 itself a total of 3477 people got killed and about 391,000 people got injured in road accidents which involved distracted drivers. Even India, in 2008, registered 130,000 deaths because of road accidents [1]. In 2013, nearly one-fifth of the crashes which hurt someone, distracted driving was involved [2]. In today's fast-paced world multitasking has become a very inextricable part of people's lives. With increasing multitasking, inattention is rising as a new nuisance. Inattention while driving can increase the chance of getting into a motor vehicle crash. Inattentions are caused due to 2 main reasons.

- Fatigue – Feeling of sleepiness or drowsiness while driving. In this case, the driver is not in the proper mental condition to drive.
- Distraction – Here though the driver is capable of concentrating on driving but gets engaged in other activities.

Many of these accidents could have been prevented had the driver been warned the moment he got distracted. In this paper, we try to explore the possibilities of training a system that can detect inattention among drivers. Such trained models can be further used to build up a driver state detection system which will effectively monitor the state of the distraction in a driver while he is driving. It is very common for people to be occupied in activities like drinking, texting each other, talking on a phone or even applying makeup while driving their vehicle. It is very important that the driver is aware of hazards that arise out of performing any of these activities while driving. It is also the responsibility of automobile manufactures to provide intelligent systems in their vehicles that detect driver distraction. A system within the vehicle if raises an alert signal every time the driver gets distracted, will help in reducing the risks that arise out of performing these activities.

Detecting distracted driver from images obtained from the dashboard camera itself is one of the most exciting applications of image recognition. This paper implements a classifier based on Convolutional Neural Networks [3] to detect distraction among the driver and also identify the cause of distraction. Identifying the category of the distraction is significant as the degree of risk associated with different distraction is dissimilar and hence need to be dealt accordingly. An image obtained from the dashboard of the vehicle is fed into the neural network and the corresponding category is recognized.

In this paper we also make an attempt to reduce the training time of the Convolutional Neural Networks [3] while making sure that accuracy of the classifier is not diminished. In our attempt to reduce the training time, we experiment with pre-trained weights that were obtained by training Convolutional Neural Network [3] models on the ImageNet [4] dataset.

This paper is organized as follows: Section 2 discusses about the other recent existing works that attempt to solve this problem. Section 3 describes the Experimental design for construction of a model. Section 4 briefly presents a description of the various VGG architectures used in this work. In Section 5, the experimental results are presented along with the analysis of efficacy of the model, and are followed by the conclusion section.

2. Literature Review

This section presents a review of some of the significant and relevant work done to solve this problem. In the work done by Céline Craye and Fakhri Karray [5] data collection was done by using a driving simulator. With the help of 8 participants where each participant contributed to an hour of video, a total of 8 hours of driving sequences was collected by them. For choosing features, they used active sensor Kinect. Active sensor Kinect can analyze color and depth data to extract features like eye behavior, arm position, head orientation and facial expressions. They created two classifiers, one for detecting distracted driver and the other for distraction recognition. For classification tasks two models, AdaBoost[6] and Hidden Markov Model[7] were trained. For detecting distraction, both models provided very similar results. Distraction was detected with the accuracy of 90% and for the type of distraction 85% was recorded.

Matti Kutila[8] worked on detecting visual and cognitive distraction detection. The dataset was obtained from a Volvo Truck and a SEAT car. For truck data, 12 professionals were hired and for car data 3 normal drivers were picked. For extracting features, stereo vision and other driver performance data such as lane tracking were utilized. Rule-based algorithm and SVM [9] or Support Vector Machine were applied on the dataset. Satisfactory results were accomplished as 80% accuracy in detecting visual distraction, and 68-86% accuracy in recognition of cognitive distraction was recorded.

Yulan Liang[10] and their team collected eye movement and driving performance data by instructing some of the participants to indulge in distracting activities such as tracking stock prices. For this, six 15 minutes drives took place, in which four had in-vehicle devices while the remaining two were baseline drives. Hence a total of 90 minutes of data in the form of video was collected. Eye movements and driving data were captured by using Seeing Machines faceLAB[11] and simulator at a rate of 60 Hz, respectively. With the application of Bayesian networks [12], nearly 80% accuracy in detecting cognitive distraction was achieved.

Ralph Oyini Mbouna[13] examined driver alertness by analyzing Head position and eye state. Data collection was done from a mounted dashboard camera. Five subjects were recruited, and each subject contributed 15 minutes of video.
Therefore, a total of 75 minutes of video with resolution 640 X 480 pixels and frame rate 30 fps was gathered. Visual parameters like eye index, head position and eye pupils were used. After extraction, these features were fed into Support Vector Machines (SVM)[9] for binary classification of

categories alert and non-alert driver. Based on the values of true positives, true negatives, false positives and false negatives, the accuracy observed for classification was 91%.

The comprehensive analysis of the above work establishes that none of the papers above applied Neural Networks. Also, [8] [10] [13] focused only on detecting distraction rather than recognizing the type of distraction. In [8], [10] faceLAB [11] of Seeing Machine was employed and [4] took the usage of Microsoft Kinect. In this work, only images recorded from the dashboard cam have been used. This work shows the application of simpler and yet very powerful model, Convolutional Neural Networks [3] which is motivated by the arrangement of neurons inside the animal visual cortex [14].

3. Experiment Design

For the purpose of this work, we used the dataset prepared by StateFarm [15]. The dataset consists of 22424 images of people either driving safely or performing any of the nine kinds of distracted behaviors described in section 3.1. Each image is a 640x480 RGB image. Out of the dataset 75% images are used for training, and the remaining 25% images are used for testing. A discussion of each of distraction category can be found below.

One of the major advantages of this dataset is that it contains subjects from different ethnic backgrounds such as Asian, Western, African, etc. Due to this, the model obtained as a result of this work is fit to perform the required tasks irrespective of the ethnic background of the driver.

3.1 Driver behavior classification

A total of 10 classes are used for classifying driver behavior, as these were identified as the most common activities performed which lead to distraction while driving. Out of the 10 classes, 1st class indicates safe driving while the remaining 9 classes indicate distraction while driving.

The 10 classes are as shown below.

- Safe Driving - Safe Driving as shown in Figure 1 is defined as the act of having hands on the steering wheel and looking in front while driving. The driver also must not engage in any of the activities mentioned below.

Figure 1. Safe Driving

- Texting using right hand while driving - If the driver is engaged in texting on his phone with his right hand while driving then the act will be classified in this category. The act is shown in Figure 2.

Figure 2. Texting using Right Hand

- Talking on Phone using right hand while driving - Performing the activity of making a call with the right hand as shown in Figure 3 while driving is also a class in the distraction category.

Figure 3. Calling using right hand

- Texting using left hand while driving –In this category the driver is texting with his left hand while driving. The act is shown in Figure 4.

Figure 4. Texting using Left Hand

- Talking on Phone using left hand while driving –If the driver is making a call using his left hand, as indicated in the Figure 5, the driver will fall into this category.

Figure 5. Calling using Left Hand

- Operating the radio while driving – Operating the radio while driving is also identified as a possible cause of distraction. Figure 6 depicts a driver performing the mentioned task.

Figure 6. Operating the Radio

- Drinking while driving - Drinking while driving as shown in Figure 7 is another class included and is a distraction drivers frequently engage in.

Figure 7. Drinking while Driving

- Reaching Behind while driving – In this category the driver turns his back towards the steering wheel as shown in Figure 8 and reaches the back seat of the car while driving.

Figure 8. Reaching Behind

- Makeup while driving - Applying makeup is another category that has been included. This act is shown in Figure 9.

Figure 9. Doing Makeup while Driving

- Talking to Passenger – In this category again the driver turns his gaze away from the windshield and looks towards his fellow passenger as shown in Figure 10. Taking his eyes off the road classifies this action as a distracting task.

Figure 10. Taking to Passenger

The classes selected are a common cause of distraction among drivers and can even lead to accidents and loss of life and property. In fact some of these activities, such as making a call while driving, are a crime in some countries for which a person can be booked if caught.

3.2 Image Augmentation and Resize

The image dataset created by StateFarm [15] was used for our experiment. In the dataset, each image is an RGB image of size 640x480 pixels. The input layer of both VGG16 and VGG19 takes as input an image of size 3x224x224. Therefore the first step was to read and resize each image to the size of 224x224 pixels.

Deep neural networks require diversified training dataset to train the model effectively. To combat this challenge we used the technique of data augmentation. For each epoch a new set of images were generated by generating various alterations on each image. Maximum ranges for degrees of shear, zoom, horizontal and vertical shifting were specified. Then random values within the maximum range for each of the mentioned parameters were applied on the image, thus generating new images but conserving the class or the category of the image.

Real-time image augmentation not only helped to provide a varied dataset and prevent the classifier from over fitting, but also helped in developing a more robust classifier which could classify much more efficiently even if some changes are brought in owing to the fact that different camera modules might be used and also there might be a shift in the position of the device.

4. Convolutional Neural Network

The idea of Convolutional neural network (CNN) [3] was proposed by LeCun [3], and has made a breakthrough in the field of image classification and target detection. Deep CNN’s introduce a significant number of hidden layers, thus reducing the dimensionality of the image and enabling the model to extract sparse image features in low dimensional space.

The conventional models consist of 2 modules – feature extraction and classification module. Because of this separation the extractor module is able to extract only a certain set of predominate features based on the algorithm used and is unable to extract the discriminating features among different categories of images. These features are then used by the classification module to classify the images.

Convolutional neural network [3] consists of multiple stages of processing layers which are able to extract these discriminating features and then these features are transferred to the classification layer to get the category of classification.

The CNN models are implemented using Keras [16] API with Tensorflow [17] in the backend. Four models were used for training the dataset. These models are inspired from VGG16 [18] and VGG19 [18].

4.1 Image Preprocessing

The mean value of RGB over all pixels was subtracted from each pixel value. Subtracting the mean value of the dataset serves to center the data. The reason mean subtraction is done is that training our model involves multiplying weights and adding biases to the initial inputs in order to cause activations which are then backpropagated [19] with the gradients to train the model.

It is important that each feature has a similar range, to prevent the gradients from getting out of specific ranges. Also, CNNs involve sharing of parameters and if the inputs are not scaled to have values of similar range, the sharing will not happen easily. This is due to the fact that one part of the image will have a large value of weights while the other will end up with smaller values.

4.2 VGG16 Model

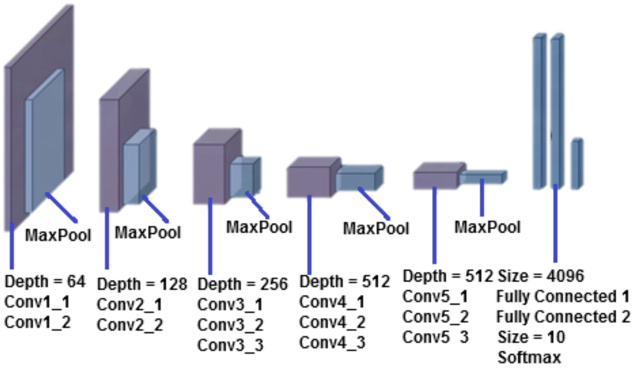


Figure 11. VGG16 Architecture [18]

VGG 16 [18] is a deep convolutional neural network model which was proposed by K. Simonyan and A. Zisserman in the paper "Very Deep Convolutional Networks for Large-Scale Image Recognition". The model was able to achieve 92.7% top-5 test accuracy in ImageNet [4].

The macro architecture of VGG16 [18] can be seen in Figure 11. The input to the ConvNets is a RGB image of size 224 × 224. The image is then passed through a stack of convolutional layers, where filters with a very small receptive field of 3 × 3 are used. The convolution stride is fixed to 1 pixel. The spatial padding of convolution layer input is 1 pixel for 3 × 3 convolutional layers thus preserving the spatial resolution. Spatial pooling has been carried out using five max-pooling layers, which follow some of the convolutional layers but not all the convolutional layers. Max-pooling is performed with a window size of 2 × 2 pixels and stride-size of 2.

The stack of convolutional layers is followed by three Fully-Connected layers: the first two layers have 4096 channels each, and the third layer performs 1000-way ILSVRC classification. The final layer is the softmax layer. All hidden layers are equipped with the rectification nonlinearity [20].

Pre-trained weights which were obtained by training the VGG [18] model on the ImageNet [4] database are used to initialize the weights in one of the models. In the other model, random initialization of weights is used. However, the original model contained 1000 channels signifying the 1000 categories which it aimed to classify. But here only 10 classes are being targeted. Therefore the final layer was popped and replaced with a fully connected softmax layer with 10 channels to perform the 10-way classifications. The remaining model was retained as it is.

The training was carried out using stochastic gradient descent [21]. A batch size of 32 and a momentum value of 0.9 were used. The learning rate was initialized with 10^{-3} and the decay rate was set to 10^{-6}. The learning was stopped after 10 epochs when pre-trained weights from ImageNet [4] were used. In the other experiment where the network weights were randomly initialized, the network was trained for a total of 50 epochs. It was observed that in spite of being a very deep model, the model with pre-trained weights required very few epochs to converge. ImageNet [4] weights can be used because of the large size of its database which contains around 1.2 million images.

4.3 VGG19 Model

VGG19 [18] model is very similar to VGG16 [18]. The VGG19 [18] architecture, as the name suggests is a 19 layer network. A representation of the architecture is shown in Figure 12.

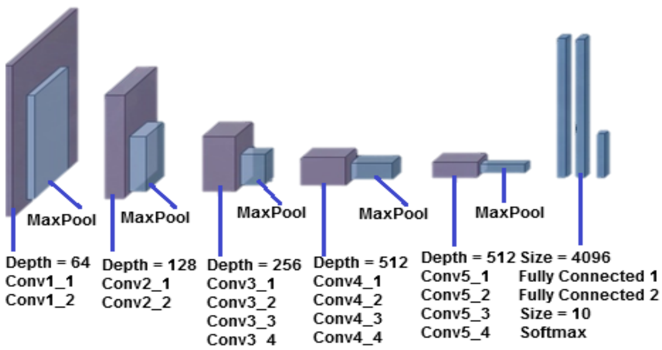


Figure 12. VGG19 Architecture [18]

In VGG19 [18], three extra convolutional layers are added to the stack. Of the three CNN layers introduced one has depth 256, while the other two layers have depth of 512 each. There is no change in the remaining fully connected layers of the network.

Here also we used a similar strategy as used in the above model. One model was initialized with pre-trained weights obtained by training the model on ImageNet [4] dataset, and the other one was initialized with random weights. Then the final layer was popped and replaced with a fully connected softmax layer with 10 channels, retaining the remaining structure. The training is carried out with similar parameters as used above.

Batch size of 32 and momentum of 0.9 was used for the purpose of training the network. The learning rate was initialized with $10^{-3}$and the decay rate was set to 10^{-6}. As in VGG16, learning was stopped after 10 epochs when pre-trained weights from ImageNet [4] were used. The model with the random initialization of weights was trained for 50 epochs. Here as well the model with pre-trained weights converged over a very small number of epochs. However, the accuracy was less as compared to the above model i.e. VGG16 [18], despite the fact that this model is deeper and thus has a greater number of parameters.

5. Results

Once the training was completed, the test dataset was run through both the models and the results obtained were recorded. A detailed analysis of the results obtained is explained below.

Table 1. Accuracy of VGG16 with Pre-Trained Weights

Scenario	Total Samples	Correct Predictions	Incorrect Predictions	Accuracy (%)
Safe Driving	622	619	3	99.52
Texting using Right Hand	565	565	0	100
Talking on phone using Right hand	579	578	1	99.83
Texting using Left Hand	587	587	0	100
Talking on phone using Left hand	581	580	1	99.83
Operating the radio	578	572	6	98.96
Drinking	581	577	4	99.31
Reaching Behind	501	500	1	99.80
Hair and Makeup	478	476	2	99.58
Talking to Passenger	534	528	6	98.88

Average Accuracy = 99.57%

Table 2. Confusion Matrix of VGG16 with Pre-Trained Weights

	1	2	3	4	5	6	7	8	9	10
1	619	1	0	0	0	0	0	0	1	1
2	0	565	0	0	0	0	0	0	0	0
3	0	0	578	0	0	0	0	0	1	0
4	0	0	0	587	0	0	0	0	0	0
5	0	1	0	0	580	0	0	0	0	0
6	4	0	1	0	0	572	0	0	1	0
7	0	1	0	0	0	0	577	0	3	0
8	0	0	0	0	0	0	0	500	0	1
9	0	0	0	0	0	0	0	0	476	2
10	2	0	0	0	2	0	0	0	2	528

Table 1 depicts the accuracy obtained for each class. Apart from the accuracy depicted, Table 2 provides a more precise and complete metric to analyze the results of the work using a confusion matrix. Both the tables provide the metrics when VGG16 [18] model is initialized with pre-trained weights.

Table 3. Accuracy of VGG16 without Pre-Trained Weights

Scenario	Total Samples	Correct Predictions	Incorrect Predictions	Accuracy (%)
Safe Driving	622	619	3	99.52
Texting using Right Hand	565	564	1	99.82
Talking on phone using Right hand	579	577	2	99.65
Texting using Left Hand	587	587	0	100
Talking on phone using Left hand	581	579	2	99.65
Operating the radio	578	574	4	99.31
Drinking	581	580	1	99.83
Reaching Behind	501	499	2	99.60
Hair and Makeup	478	471	7	98.54
Talking to Passenger	534	524	10	98.13

Average Accuracy = 99.43%

Table 4. Confusion Matrix of VGG16 without Pre-Trained Weights

	1	2	3	4	5	6	7	8	9	10
1	619	1	0	0	0	0	0	0	1	1
2	0	564	0	0	0	0	1	0	0	0
3	0	0	577	0	0	0	1	0	1	0
4	0	0	0	587	0	0	0	0	0	0
5	0	0	0	0	579	0	1	0	1	0
6	1	0	0	1	0	574	0	0	1	0
7	0	0	0	0	0	0	580	0	1	0
8	0	0	0	0	0	0	0	499	1	1
9	0	0	2	0	0	0	3	0	471	2

10	3	1	0	0	1	0	0	0	5	524

Table 3 and Table 4 are obtained when the weights of VGG16 [18] model are not initialized with pre-trained weights but are initialized randomly. Here again, the Table 3 depicts accuracy and Table 4 is the confusion matrix.

Table 5. Accuracy for VGG19 with Pre-Trained Weights Results

Scenario	Total Samples	Correct Predictions	Incorrect Predictions	Accuracy (%)
Safe Driving	622	615	7	98.87
Texting using Right Hand	565	564	1	99.82
Talking on phone using Right hand	579	578	1	99.83
Texting using Left Hand	587	586	1	99.83
Talking on phone using Left hand	581	578	3	99.48
Operating the radio	578	573	5	99.13
Drinking	581	581	0	100
Reaching Behind	501	499	2	99.60
Hair and Makeup	478	446	32	93.30
Talking to Passenger	534	529	5	99.06

Average Accuracy = 98.98%

Table 6. Confusion Matrix of VGG19with Pre-Trained Weights

	1	2	3	4	5	6	7	8	9	10
1	615	1	0	0	1	0	1	0	0	4
2	0	564	0	0	0	0	1	0	0	0
3	0	0	578	0	0	0	1	0	0	0
4	0	0	1	586	0	0	0	0	0	0
5	1	0	0	0	578	0	2	0	0	0
6	3	0	1	0	0	573	0	1	0	0
7	0	0	0	0	0	0	581	0	0	0
8	0	0	1	0	0	0	0	499	0	1
9	0	1	10	0	0	0	14	2	446	5
10	1	1	0	0	1	0	0	0	2	529

Now VGG19 [18] model is used for training. Here again, it is initially initialized with pre-trained weights from ImageNet [4]. Table 5 depicts the accuracy obtained for each class and Table 6 is the confusion matrix that is obtained from the model.

Table 7. Accuracy of VGG19 without Pre-Trained Weights

Scenario	Total Samples	Correct Predictions	Incorrect Predictions	Accuracy (%)
Safe Driving	622	618	4	99.36
Texting using Right Hand	565	564	1	99.82
Talking on phone using Right hand	579	578	1	99.83
Texting using Left Hand	587	585	2	99.66
Talking on phone using Left hand	581	580	1	99.83
Operating the radio	578	573	5	99.13
Drinking	581	581	0	100
Reaching Behind	501	500	1	99.80
Hair and Makeup	478	470	8	98.33
Talking to Passenger	534	523	11	97.94

Average Accuracy = 99.39%

Table 7 and Table 8 are obtained for VGG19 [18] model when again the weights are initialized randomly instead of using pre-trained weights.

Table 8. Confusion Matrix of VGG19without Pre-Trained Weights

	1	2	3	4	5	6	7	8	9	10
1	618	1	0	0	0	0	0	0	1	2
2	0	564	0	0	0	0	1	0	0	0
3	0	0	578	0	0	0	0	0	1	0
4	1	0	0	585	0	0	0	0	1	0
5	0	0	0	0	580	0	1	0	0	0
6	2	0	0	0	0	573	0	0	3	0
7	0	0	0	0	0	0	581	0	0	0
8	0	0	0	0	0	0	1	500	0	0
9	1	1	0	0	0	2	4	0	470	2
10	2	1	0	0	1	1	0	0	6	523

During the experiments, it was clearly observed that the use of pre-trained weights significantly reduced the training time, however this didn't lead to any loss of accuracy of the trained network.

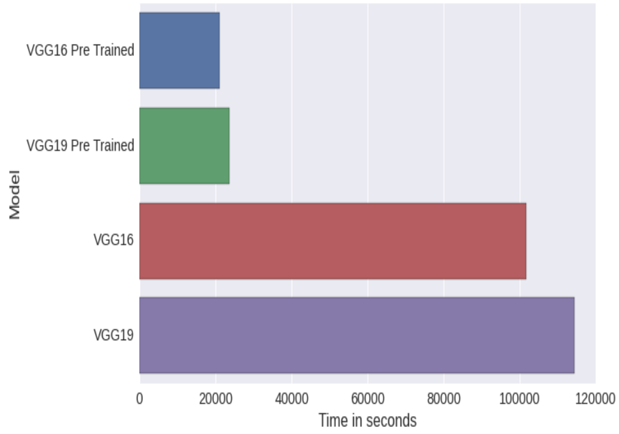


Figure 13. Training Time Comparison of different models

All of the models created in this work, performed well on the test data.VGG19 [18] although being deeper than VGG16 [18] was outperformed by the latter. This is maybe due to the fact that VGG19 [18] learned some extra irrelevant features and which lead to overfitting on the training data set and hence diminishing its accuracy.

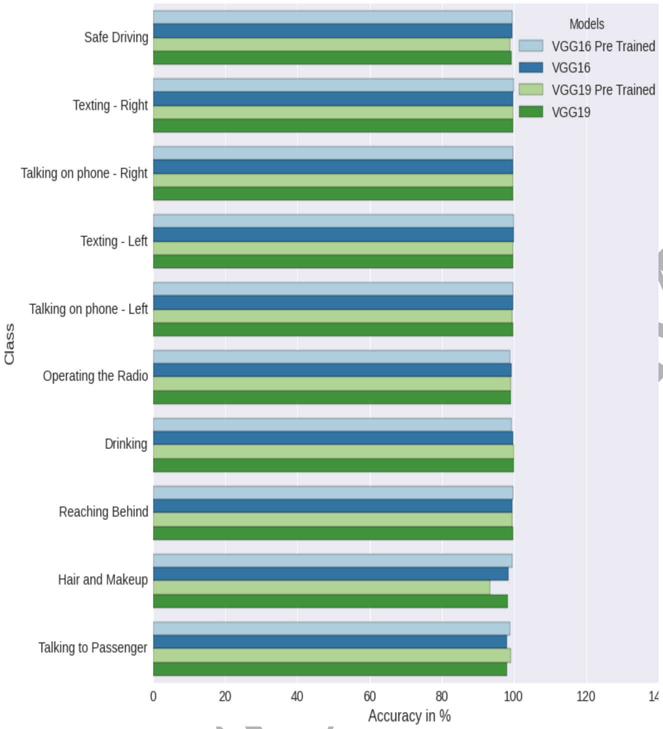


Figure 4. Accuracy comparison on results obtained while using different models

Moreover, better results were obtained if pre-trained weights from ImageNet [4] model are used for initialization of weights in the VGG16 [18] model. VGG16 [18] model with pre-trained weights gives the best accuracy at an impressive 99.57 %. Also, the models which were initialized with pre-trained weights converged much faster than the other models, resulting in reduced training time. Best and very consistent results have been shown by this VGG16 [18] model.

6. Conclusion

Deep learning using Convolutional Neural Networks [3] has become a hot area in Machine Learning research, and it has been extensively used in image classification, voice recognition, etc. In this paper, we use Deep Convolutional Networks for detecting distracted drivers and also identifying the cause of their distraction using the VGG16 [18] and VGG19 [18] model.

The above results suggest that the methods discussed in this work can be used to develop a system using which distraction while driving can be detected among drivers. The model proposed can automatically identify any of the mentioned 10 classes of distraction and identify not only basic distraction but also their cause of distraction. With an accuracy of more than 99%, the mentioned system was shown to be efficient and workable. The proposed system can be a part of some Driver State Monitoring System which will effectively monitor the state of the driver while he is driving. Driver state monitoring has been becoming increasingly popular these days and many automobile giants have started adopting such systems as a methodology to prevent accidents. These systems, when installed inside vehicles will raise warnings whenever the driver gets distracted, thus trying to prevent any accidents due to distraction from the driver.

Also in this work a significant amount of training time has been shown to be reduced. When pre-trained weights from ImageNet [4] were not used, the training time increased by around 50 times for both VGG16 [18] and VGG19 [18]. A graphical representation of time elapsed is depicted in Fig. 15. This drastic reduction in training time was achieved without diminishing the accuracy of our classification models.

In future work as an extension to this work, more categories of distraction can be brought in. Even considering certain specific scenarios, which were not targeted in the present work, such as detecting drowsiness among drivers may also provide an opportunity to widen the scale of the work and build a more efficient system.

7. Acknowledgements

The dataset for training our models was developed by State Farm Insurance Co. Thanks are due to the Company & Mr. Taylor Smith, principal data scientist at State Farm, for his support and cooperation with us on the matter of use of their dataset.

References

[1] "Mobiles phone use and increasing road accidents" Raghvan Guruswami, accessed September 10, 2016, http://blogs.siliconindia.com/guru_raghavan/Technology/Mobile-phones-use-and-increasing-road-accidents-bid-rhF4230a61127033.html

[2] "Distracted Driving", Centres for Disease Control and Prevention, accessed September 8, 2016,http://www.cdc.gov/motorvehiclesafety/distracted_driving.

[3] LeCun et al. 1998, "Gradient-based learning applied to document recognition." *Proceedings of the IEEE* 86, no. 11 2278-2324.

[4] J. Deng, W. Dong, R. Socher, L.-J, Li, K. Li and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In CVPR09, 2009

[5] Craye, Céline, and Fakhri Karray. "Driver distraction detection and recognition using RGB-D sensor." *arXiv preprint arXiv:1502.00250* (2015).

[6] Freund, Yoav, Robert Schapire, and N. Abe, 1999, "A short introduction to boosting." *Journal-Japanese Society for Artificial Intelligence* 14, no. 771-780 : 1612.

[7] S. B. Wang, et al. 2006, "Hidden conditional random fields for gesture recognition," in ComputerVision and Pattern Recognition, 2006 IEEE Computer Society Conferenceon, vol. 2. IEEE, pp. 1521–1527.

[8] Kutila, et al, 2007, "Driver distraction detection with a camera vision system." In *2007 IEEE International Conference on Image Processing*, vol. 6, pp. VI-201. IEEE.

[9] Cortes, Corinna, and Vladimir Vapnik. "Support-vector networks." *Machine learning* 20, no. 3 (1995): 273-297.

[10] Liang, Yulan, John Lee, and Michelle Reyes. "Nonintrusive detection of driver cognitive distraction in real time using Bayesian networks." *Transportation Research Record: Journal of the Transportation Research Board* 2018 (2007): 1-8.

[11] "faceLAB", Eyecomtec accessed September 17 2016 https://www.eyecomtec.com/3132-faceLAB

[12] Li, X., and Q. Ji, 2005, "Active Affective State Detection and User Assistance with Dynamic Bayesian Networks", IEEE Transactions on Systems, Man, and Cybernetics—Part A: Systems and Humans, Vol. 35, No. 1, pp. 93–105.

[13] Mbouna et al. 2013, "Visual analysis of eye state and head pose for driver alertness monitoring." *IEEE transactions on intelligent transportation systems* 14, no. 3: 1462-1469.

[14] Hubel et al. 1968, "Receptive fields and functional architecture of monkey striate cortex." *The Journal of physiology* 195, no. 1: 215-243.

[15] State Farm Distracted Drivers Dataset, accessed on June 15 2017, https://www.kaggle.com/c/state-farm-distracted-driver-detection/data

[16] Chollet, François. "Keras." (2015). Accessed on June 15 2017, https://github.com/fchollet/keras.

[17] Abadi, Martín, et al. 2016, "Tensorflow: Large-scale machine learning on heterogeneous distributed systems." *arXiv preprint arXiv:1603.04467*.

[18] Simonyan, Karen, and Andrew Zisserman, 2014 "Very deep convolutional networks for large-scale image recognition." *arXiv preprint arXiv:1409.1556*.

[19] Rumelhart et al. 1988, "Learning representations by back-propagating errors." *Cognitive modeling* 5, no. 3: 1.

[20] Hahnloser et al., 2000, "Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit." *Nature* 405, no. 6789 : 947-951.

[21] Bottou, Léon, 2010, "Large-scale machine learning with stochastic gradient descent." In *Proceedings of COMPSTAT'2010*, pp. 177-186. Physica-Verlag HD.

[22] NHTSA, "Distraction.gov, official us government website for distracted driving", accessed on June 15 2017, https://www.nhtsa.gov/risky-driving/distracted-driving