

End-to-End Deep Learning for Driver Distraction Recognition

Arief Koesdwiady^(✉), Safaa M. Bedawi, Chaojie Ou, and Fakhri Karray

Center for Pattern Analysis and Machine Intelligence, University of Waterloo,
Waterloo, ON, Canada

{abkoesdw,sbedawi,c9ou,karray}@uwaterloo.ca

Abstract. In this paper, an end-to-end deep learning solution for driver distraction recognition is presented. In the proposed framework, the features from pre-trained convolutional neural networks VGG-19 are extracted. Despite the variation in illumination conditions, camera position, driver's ethnicity, and genders in our dataset, our best fine-tuned model, VGG-19 has achieved the highest test accuracy of 95% and an average accuracy of 80% per class. The model is tested with leave-one-driver-out cross validation method to ensure generalization. The results show that our framework avoided the overfitting problem which typically occurs in low-variance datasets. A comparison between our framework with the state-of-the-art XGboost shows that the proposed approach outperforms XGBoost in accuracy by approximately 7%.

Keywords: Driver distraction · Deep learning · Intelligent transportation system

1 Introduction

Distracted driving is much more dangerous than most people realize. According to the most recent published World Health Organization (WHO) report, it was estimated that, in 2015, over 1.25 million people were killed on the roads worldwide, making road traffic injuries a leading cause of death globally [6]. Driver errors still remain the main cause of accidents in the roads. Using the cellphone for texting, talking and navigation as well as drowsiness are different types of activities that drastically decrease drivers attention to the road. In this background, research has focused on help to improve these alarming statistics. Research includes, but is not limited to: research on identifying driver behaviour to help the industry creating solutions to reduce the effect of driver distraction [3] and research on self-driving cars [1], where lane marking detection, path planning, and control are the area of enhancements. In [3], a review is provided on algorithms used in driver distraction detection. The main methods are focused on face detection, face/hand tracking and detection of facial landmarks [9]. Among the algorithms used are: SVM, Histogram of Oriented Gradients, Artificial Neural Networks (ANN), and Deep Neural Networks (DNN). The reported results for

these methods show that they are affected to various degrees, by illumination, skin colour and camera position inside the car.

In this paper, an end-to-end deep learning based classifier is investigated for driver distraction detection. This is motivated by its performance in self-driving car [1] and to have the system operated without direct human interaction. The purpose of this paper is to investigate the robustness of the suggested framework under different illuminations, different drivers type and different camera positions. To our best knowledge, an end-to-end deep learning framework has not been used for driver distraction detection.

The rest of this paper is organized as follows. An overview on deep learning framework, off-the-shelf feature extraction, dimensionality reduction and how it is used in classification are introduced in Sect. 2. In Sect. 3, a description of our dataset, experimental setup and fine-tuning approaches are described. Analysis of our results is in Sect. 4 and finally the conclusion and future work are presented in Sect. 5.

2 End-to-End Deep Learning Framework

Driver distraction recognition problem can be treated as a multi-class classification process to map the input observations to a driver state. The developed system includes three main components, as shown in Fig. 1. The first component is a variant of convolutional deep neural network for high-abstracted feature extraction. Followed by a max pooling layer which reduces the dimension of features. The last component includes 6 fully-connected layers and a softmax layer.

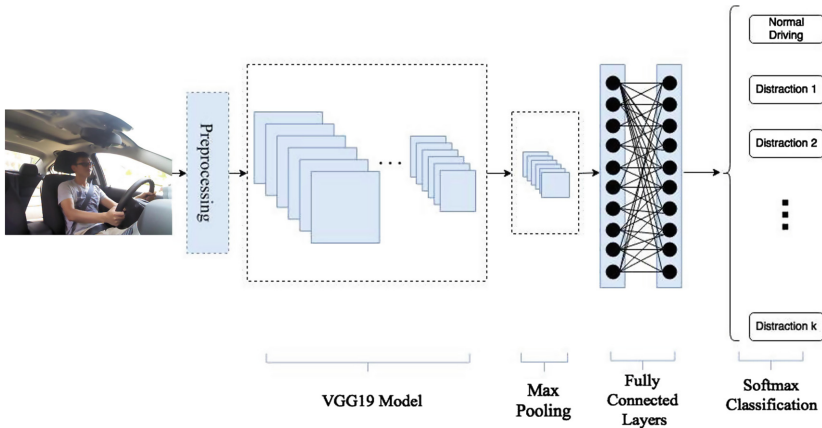


Fig. 1. End-to-end deep learning framework.

2.1 Feature Extraction

A common ConvNet is a stack of convolutional layers, pooling layers, most often followed by several fully-connected layers. The convolutional layer and pooling layer operate on small local input patches, and the combination of these two layers makes the network more robust to location variation of objects in the given images.

As a variant of ConvNet, the VGG 19 network is firstly proposed for images classification, object detection, and objects localization in competition ImageNet [7]. It is quickly accepted in many other computer vision and image processing researches, due to its simple structure and moderate number of parameters. There are two common methods to adopt this network: by fine-tuning all parameters in VGG and extracting highly-abstracted features for pre-trained VGG models. The research work in this paper follows the second method and extract represented features from the VGG19 model.

The architecture and configuration of VGG19 can be found in [7] is also roughly summarized here. The input should be a 224×224 RGB picture. The size of kernel is 3×3 , which makes the following layers contains small local patch information, the stride for convolution is 1. The max-pooling is performed on a 2×2 patch with stride of 2. The last three fully-connected layers in VGG19 are dropped, and the remaining structure works as a feature extractor.

2.2 Classification

The classifier in the original VGG19 is a three-layers fully-connected network, which is designed and trained for classification of images that contains different objects. The dimension of features maps after the last max-pooling layer in VGG19 is $7 \times 7 \times 512$. To reduce the dimension of features and speed up the learning process, another max-pooling layer is connected to the last pooling layer in the VGG19 model and before the DNN classifier. The max-pooling is also performed over a 2×2 pixel window, with stride 2. In this work, the XGBoost and a six-layers fully-connected networks are exploited as classifiers to classify distracted drivers. The classification results of this two classifiers is presented and compared in Sect. 4.

The fully-connected network classifier contains 6 layers with 1000 nodes in each layer, and it is trained by back-propagation with stochastic gradient descent optimization. The XGBoost is a fast-implementation of gradient boosting trees [2, 4]. Many successful solutions in Kaggle competitions are developed with this additional tree boosting method. However, the learning process for gradient boosting tree is time-consuming, thus it is not suitable for working directly on pixel level of large images.

3 Experiments

This section presents descriptions of the dataset gathered using a video camera, data pre-processing, and experimental setups.

3.1 Dataset

Taking into account the comfort of the drivers, a video camera is used in this work to capture driver situations. The camera is located such that the upper-body, hand positions, and rear-part of the car are captured and available to analyze. From the camera, sets of 640×480 - RGB video images with a frame rate 15 frames per second are obtained. In this experiment, two different cars are used in different lighting conditions. This way, the proposed system is forced to learn real-world driving situations. These situations are illustrated in Fig. 2.

As can be seen in Fig. 2, the experiment is carried out by participants from different ages, genders, and ethnicity. Ten drivers were involved in the experiments. Each driver was asked to perform or mimick the following driving activities: Normal/safe driving, Text messaging using left and right hand, Phone calling using left and right hand, Operating radio/navigation systems, Reaching objects at the rear-part of the car and Drinking using left or and right hand.



Fig. 2. Distraction types (from top-left to right-bottom): normal driving, text messaging (right-hand), drinking, reaching object at the back, calling on the phone, operating radio.

3.2 Experimental Settings

In this work, the driving activities are grouped into three types: distraction involving left hand, right hand, and distraction while reaching object at the back of the car. Together with normal driving, 4 classes of driving states are selected. For simplicity, these 4 classes will be called normal driving, distracted left, distracted right, and distracted back throughout the paper. As grouping the classes has caused imbalance in our data set, we increased the number of images for normal driving cars by flipping the images vertically and horizontally and sharpening the images to avoid imbalance classification problems. The images generated by the camera are reshaped to 224×224 pixels so that it can be fed in the feature extraction module VGG19. Subsequently, a pre-processing stage involving mean subtraction of RGB values was implemented.

After the features are extracted using VGG19 model, a max-pooling layer is implemented for dimensionality reduction. Furthermore, fully-connected layers are stacked for the classifications using a soft-max layer. The fully-connected layers are trained using back-propagation with stochastic gradient descent optimization. To obtain the best performance, the parameter tuning was done by varying the number of hidden layers and neurons. However, the number of neurons was kept fixed throughout the hidden layers. For example, six hidden layers with each layer consists of 1000 neurons.

The most popular non-linear activation function, namely rectified linear unit (ReLU), is implemented so that the deep neural network can learn much faster without unsupervised pre-training and, in the same time, avoid the vanishing gradient problem [5]. Moreover, to avoid over-fitting during the training, the drop-out method is introduced [8]. The training of the deep neural networks in this work uses a computing system that is powered by Intel Core i5 Quad Core 3.5 Ghz, 16 GB of RAM, and a GTX1070 8 GB GPU card.

To analyze the performance of the proposed model, XGBoost [2] method is used for comparison. XGBoost is a machine learning algorithm that has recently been winning major machine learning competitions such as Kaggle. This method is selected so that the state-of-the-art algorithm, i.e., deep learning, is fairly compared with another state-of-the-art non-connectionist machine learning algorithm. This method is implemented for classification after the features are extracted using the VGG19 model. The parameters of the model such as number of rounds, maximum depth of the tree, minimum child weight, and β (minimum loss reduction required to make a further partition on a leaf node of the tree) are tuned using similar cross-validation procedure as the proposed deep neural network.

The performance of both models are measured based on classification accuracy and multi-class log-loss. These metrics were applied to 1 test driver after the model is trained using the other 9 drivers. This scheme was applied for all drivers and is also known as leave-one-subject-out cross validation to ensure the model to generalize under different types of drivers.

4 Results and Analysis

Table 1 shows the performance of two different classifiers on each class. The results shows that the DNN classifier is dominating XGBoost on three classes (Normal Driving, Distraction Right, and Distraction Back); while on Distraction Left class, the difference is minimal. The performance of these two classifiers on the average precision is more close than that on recall and F1-measure.

In addition, the class of Distraction Left is well discriminated for both classifiers, the F1-Measure of both classifiers are almost 1. The table also shows that the precision for Distraction Right and Normal Driving classes are larger than their recall values, while the opposite can be found for Distraction Back. This difference shows that the system classifies more samples into Distraction Back, not the opposite.

Table 1. Per class performance comparison between DNN and XGBoost

Class	DNN			XGBoost		
	Precision	Recall	F1-Measure	Precision	Recall	F1-Measure
Normal driving	0.96	0.65	0.75	0.94	0.67	0.76
Distraction left	0.99	0.99	0.99	1.00	1.00	1.00
Distraction right	0.80	0.71	0.70	0.64	0.59	0.54
Distraction back	0.68	0.85	0.73	0.59	0.75	0.61
Average	0.86	0.80	0.80	0.79	0.75	0.73

The performance of both classifiers on different drivers is presented in Table 2. As can be seen, the accuracy on different drivers by DNN classifier changes from 68.88% to 98.58%; while in XGBoost, it changes in a range of 64.25% to 85.38%. This significant disparity on different drivers shows that the learning process is limited by the magnitude of the dataset, and pictures of the distracted driver are related.

The variance of accuracy of DNN is 0.101, and for XGBoost it is 0.0635. The smaller variance shows that XGBoost is more stable but the average accuracy is 75.04%, also the standard deviation of Log-Loss by XGBoost is larger than that of DNN. The receiver operating characteristic curve (ROC) by changing the minimum probability of accepting the result is illustrated in Fig. 3. The Fig. 3(a) is based on the testing result of driver 3, while Fig. 3(b) is based on driver 8. The ROC for the class of Distraction Left in Fig. 3(b) reveals that by changing acceptance probability it is possible to achieve a good accuracy on Distraction Left class.

Table 2. Driver performance comparison between DNN and XGBoost

Driver	DNN		XGBoost	
	Accuracy	Log-loss	Accuracy	Log-loss
1	94.63	0.740	82.08	0.395
2	80.75	0.619	76.89	0.547
3	98.58	0.602	73.80	0.816
4	76.92	0.609	85.38	0.459
5	69.79	0.674	73.71	0.791
6	85.38	0.633	76.63	0.647
7	79.21	0.758	78.92	0.606
8	68.88	0.911	64.25	0.784
9	70.63	0.588	70.50	0.595
10	77.83	0.683	68.29	0.911
Average	80.258	0.682	75.04	0.655
Standard deviation	10.10	0.10	6.35	0.17

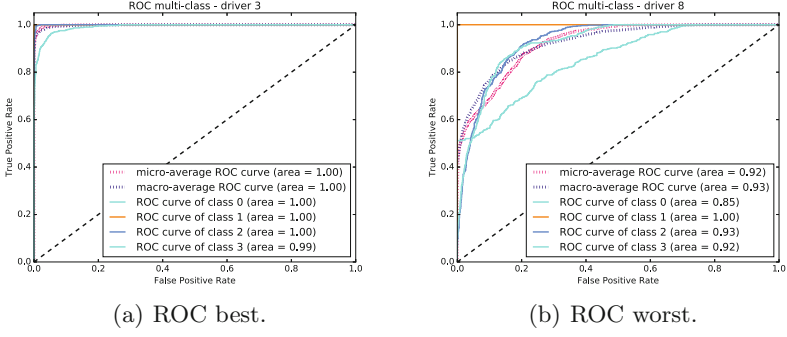


Fig. 3. ROC of results

The reason for the difference in performance of different classes can be found by checking the position between the camera and the driver. The camera is mounted on the dash board before the assistant's seat, and it takes pictures of the driver from the side view, so the activity of the hand close to the camera is represented more clearly than the activity of the other hand. By changing the position of the camera, the occlusion between two hands in images can be mitigated and classification accuracy of other classes can be improved.

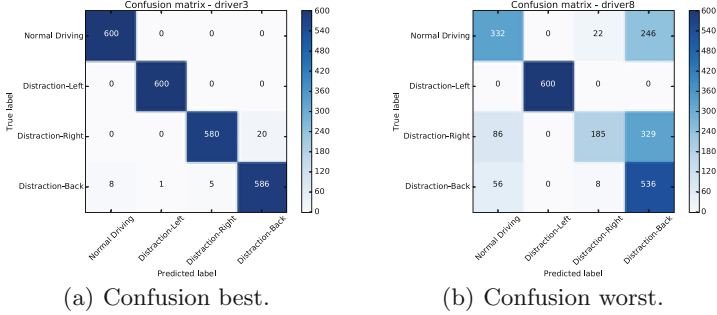


Fig. 4. Confusion matrices for different drivers

Figure 4 are confusion matrices of the best testing result (Driver 3) and the worst result (Driver 8). Figure 4(b) shows clearly that many samples of Normal Driving and Distraction Left are classified as Distraction Back. This result means the Normal Driving and Distraction Back share many similar states or images. This is also verified by checking the data.

5 Conclusion

In this paper, an end-to-end deep learning solution for driver distraction recognition is suggested in which the pre-trained convolutional neural networks

VGG-19 are used. Despite the challenging aspects considered in the dataset in terms of different illumination conditions, camera positions and variations in driver’s ethnicity, and genders, the proposed end-to-end framework was able to detect different classes with a best test accuracy of 95% and an average accuracy of 80% per class. It also outperformed XGBoost by 7% classification accuracy. The main challenge of end-to-end framework comes from the difficulty of tuning the neural networks as it requires significant amount of resources and time. Future work will include increasing the size of the dataset and using ensembles to boost in the accuracy.

Acknowledgment. The authors would like to thank Dr. Alaa Khamis from Suez University, Egypt for his generous assistance with the data collection process.

References

1. Bojarski, M., Del Testa, D., Dworakowski, D., Firner, B., Flepp, B., Goyal, P., Jackel, L.D., Monfort, M., Muller, U., Zhang, J., et al.: End to end learning for self-driving cars. arXiv preprint [arXiv:1604.07316](https://arxiv.org/abs/1604.07316) (2016)
2. Chen, T., Guestrin, C.: XGBoost: a scalable tree boosting system. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 785–794. ACM (2016)
3. Fernández, A., Usamentiaga, R., Carús, J.L., Casado, R.: Driver distraction using visual-based sensors and algorithms. *Sensors* **16**(11), 1805 (2016)
4. Friedman, J.H.: Greedy function approximation: a gradient boosting machine. *Ann. Stat.* 1189–1232 (2001)
5. Glorot, X., Bordes, A., Bengio, Y.: Deep sparse rectifier neural networks. In: *Aistats*, vol. 15, p. 275 (2011)
6. National Center for Statistics and Analysis. Distracted driving 2013. Technical report, The National Highway Traffic Safety Administration (2015)
7. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint [arXiv:1409.1556](https://arxiv.org/abs/1409.1556) (2014)
8. Srivastava, N., Hinton, G.E., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: a simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* **15**(1), 1929–1958 (2014)
9. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2001, vol. 1, p. I. IEEE (2001)

Image Analysis and Recognition

14th International Conference, ICIAR 2017, Montreal,

QC, Canada, July 5-7, 2017, Proceedings

Karray, F.; Campilho, A.; Cheriet, F. (Eds.)

2017, XVII, 673 p. 293 illus., Softcover

ISBN: 978-3-319-59875-8