IOSR Journal of Computer Engineering (IOSR-JCE)
ISSN: 2278-0661, p-ISSN: 2278-8727, Volume 23, Issue 3, Ser. I(May – June 2021), PP 24-30
www.iosrjournals.org

A Deep Learning Approach to Driver Fatigue Detection via Mouth State Analyses and Yawning Detection

Burcu Kır Savaş[1], Yaşar Becerikli[1]
[1](*Computer Engineering Department, Kocaeli University, Turkey)*

Abstract: *This paper aims to examine the effect of fatigue caused by temporary/permanent physical characteristics on driver behavior and detect fatigue drivers as a result of the research. In this context, the driver was monitored by fixed cameras in daylight conditions. Driver fatigue detection system was developed to use yawning for fatigue detection and classify this characteristic with the Convolutional Neural Network (ConNN) model. Proposed method is trained and evaluated on YawDD, Nthu-DDD and KouBM-DFD dataset. Experimental results demonstrate that the proposed ConNN can efficiently detect driver fatigue status with driving images. The proposed ConNN algorithm accuracy is higher than other CNN-based methods. Accuracy rates shows 99.35%, 99.25% and 99,01%, respectively.*
Key Word*: Human Computer Interaction, Behavioral-based measurement, Driver fatigue, Mouth detection, ConNN.*

Date of Submission: 08-05-2021 Date of Acceptance: 23-05-2021

I. Introduction

Considering the current state of today's technologies, physiological-based, vehicle-based and behavioral-based measurement ideas are focused to detect driver fatigue. Physiologically based measurements rely on signals such as brain waves, heart rate and respiratory rate [1], [2] to detect driver fatigue. Physiological signals often require physical contact with the driver, which can cause discomfort to the driver. Vehicle-based measurements enable the determination of fatigue based on the use of vehicle equipment (the driver's steering grip, the power applied to the pedal, etc.) [3], [4]. The problem is driver fatigue detection works in limited situations because the vehicle steering is used as input [5] and these systems are very dependent on the road characteristic. The reliability, accuracy and sensitivity of these systems which are on the market still remain uncertain [6].

These shortcomings paved the way for the development of a third approach, behavioral-based measurement systems [7],[8]. In behavioral-based measurements, driver fatigue is detected by focusing on the driver's behavior with computer vision and image processing approaches. In these measurements, focal point are driver's face, eyes, mouth and head state changes. Behavioral-based fatigue detection methods can be applied without distracting the driver. To determine and evaluate these characteristics on images of the driver instantly, it is determined the current status of the driver and whether he is tired or not [9], [10], [11]. Many researchers have proposed that yawning which is considered one of the behavioral-based fatigue detection conditions can be considered as a physiological state to determine whether the driver fatigued or unfatigued [12] [13], [14], [15].

Zhang et al. (2015) [16] proposed a yawn detection system consisting of a face, nose and yawn detector in their study. They have developed a nose tracking algorithm by combining Kalman filter with a special open-source Tracking, Learning and Detection method. Finally, features such as nose tracking confidence value, gradient features around the corners of the mouth, and facial movement features were classified with the CNN model. Lai et al. (2015) [17] used deep learning methods to detect facial cue points. CNN and Recursive Neural Networks (RNN) methods were used to determine facial cues. Yan et al. (2016) [18] used the CNN model to automatically learn and predict predefined driving postures in their study. In their studies, the authors first trained and then classified in an unsupervised learning method called sparse filtering with the CNN model. In the study, the Southeast University driving dataset was used for classification. Gu et al. (2018) [19] proposed a multi-purpose hierarchical CNN model for the fatigue detection system, and PERCLOS and FOM criteria were used. Zhao et al. (2020) [20] used MTCNN architecture for face detection and feature point detection in their proposed driver fatigue detection algorithm, and they proposed a convolutional neural network called EM-CNN to detect the states of the eyes and mouth from the ROI images they obtained. Autonomous cars (self-driving cars) have emerged in recent years [21]. Based on the behavioral fatigue prediction technique, auto suppliers such as Bosch [22], and NVidia [23] are developing driver fatigue detection systems.

In the study, driver fatigue is monitored in real time by using a behavioral model on drivers. We consider deep convolutional neural networks [24] because ConNNs have developed rapidly in the field of machine vision, especially for face detection [25]. First of all, the driver is monitored by fixed camera (s) in daylight conditions. Secondly, using the ConNN, the driver's face and mouth area are determined using the images obtained and the driver's yawning is classified. At the end of the classification, depending on the yawning and FOM (Flexing Frequency), the fatigue degree of the driver is decided. In this study, in addition to the YawDD [26], Nthu-DDD [27] data sets in the literature, the original data set called KouBM-DFD [28] (Kocaeli University Computer Engineering Driver Fatigue Dataset), which was prepared and added to the literature, was used to determine the fatigue degree of the driver. In this context; In order to find the most optimum result, 2 different models and their sub-models were created. Thus, we propose a vision-based system for detection of yawning is presented for detecting driver fatigue together with its implementation as a Human–Computer Interaction system.

The rest of this article is organized as follows: Section II presents a overview of the system architecture, such as the data sets used, implementation and fatigue parameter. Section III proposed ConNN model is given and describes experimental results obtained on the model. Then, Section VI gives the results of this paper.

II. Overview of the System Architecture

For this study, 3 different data sets were used. The first of this dataset is YawDD data set. This data set contain two video data sets with various facial characteristics used for mouth recognition and monitoring and this dataset contains 351 videos. The second data set is Nthu-DDD data set. It consists of different types of sleepy and non-dormant activities and different situations including day and night. The last one is KouBM-DFD data set prepared within the scope of this study at our IPCV research laboratory and added to the literature. This data set contains various facial characteristics used for eye recognition and tracking, mouth recognition and monitoring. The data set consists of a community of male and female drivers of different nationalities. In this study, during ConNN training, data sets are divided into training sets and test sets. YawDD, Nthu-DDD and KouBm-DFD data sets are used for training and test set. The distribution of these data sets is shown in Table I.

Table 1. The distribution of these data sets

Data sets	**YawDD**	**Nthu-DDD**	**KouBm-DFD**
Training set	51600	28943	26694
Test set	11710	4964	5730
Total set	62910	33907	32424

Implementation

SGD optimization algorithm was used in the training phase. In this phase, batch size of 32, learning rate is initialized as 0.01, momentum of 0.6 and weight decay of 0.0005. We proposed a cross entropy loss by monitoring the loss function. The accuracy calculation used in the study is given in (1).

$$Accuracy = \frac{1}{N}\sum_{i}^{N} 1(y_i == \hat{y}) \tag{1}$$

Here, it corresponds to the tensor of target values (data held in an array of any size) and predicted values, and the accuracy value is calculated according to the total true positive and true negative values in the binary classification complexity matrix. N value corresponds to the total numbers of data.

The experiment is coded in python language and utilized from Nvidia CUDA. Our tests were done on an NVIDIA GeForce GT540M, CUDA 1GB, Intel Core i7, 4GB DDR3, Ubuntu 16.04.

Fatigue Parameters

In this study, mouth state knowledge was used as the fatigue parameter. This situation depends on the yawning number and yawning frequency. The FOM is the ratio shown as a percentage of the driver-mouthed states to the total situation within a certain period of. FOM metric is given in (2).

$$\mathrm{N} = \frac{M_c}{M_t} \tag{2}$$

In the equation M_c represents the number of open mouth frames within a given time frame. N_t refers to the total number of open and closed mouth frames in a given time period.

III. Experimental Results

Selection of modifiable model parameters (weight, layer, kernel / kernel dimensions, kernel parameters, etc.) is one of the important steps in problem solving in mathematical model-based approaches. It is quite difficult to find the optimum parameters for the model parameters in question for any problem. By trying different model parameters many times, the most suitable parameter values are tried to be determined. For this reason, it would be useful to develop adaptive parameter changing approaches. In this study, tests were carried out to find optimum parameter values by changing the number of filters, stepping number and filling number hyper parameters in convolution layers.

In order to obtain optimum parameter values in the designed ConNN model, a total of 2 different models (Model-1 and Model 2) were created by changing the hyper parameter (the number of convolution steps). On the created Model-1, 3 different sub-models (Model-1 (a), Model-1 (b) and Model-1 (c)) were designed by changing the number of filters in the convolution layers.

Network structure of ConNN models

Proposed ConNN model height is shown “Fig. 1”. This model consists of input, 3 convolutional layers, 3 activation layers, 3 pooling layers, flatten layer and 2 fully connected layers and output.

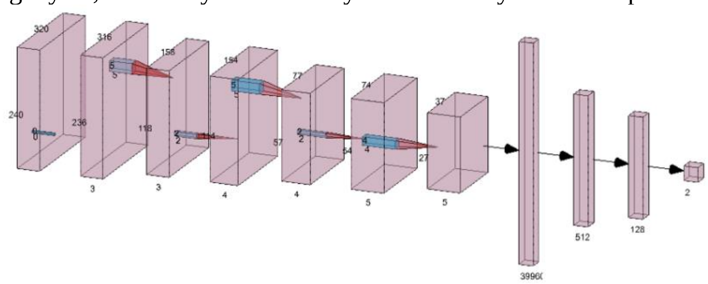
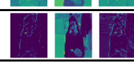
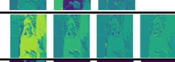
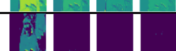
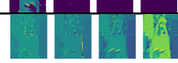
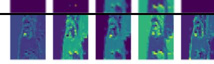
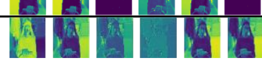
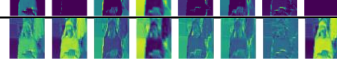
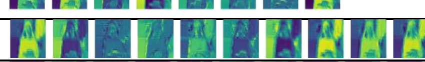
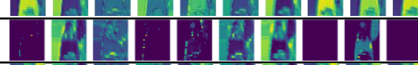
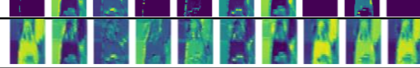
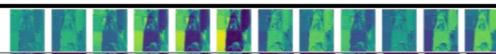
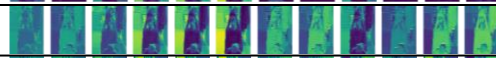
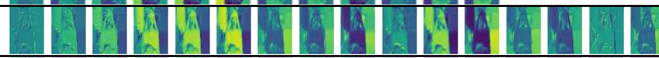
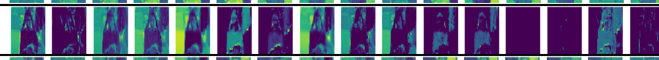
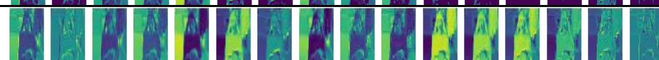
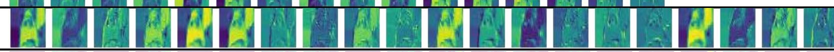
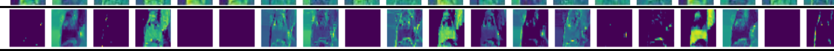
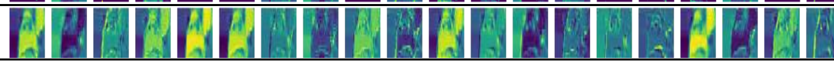
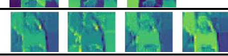
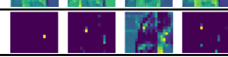
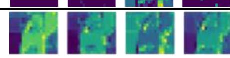
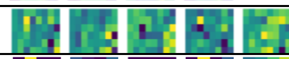


Fig. 1. Proposed ConNN model

The network structure of the designed models is given in Table 2. Hyper parameters changed in the ConNN model are highlighted in bold. For the KouBm-DFD dataset, the changes in each layer output are shown.

Table 2. The network structure of the designed models

Model	Layers	Parametres	Layer Output
Model-1 (a)	Conv1	5×5, **3**, **stride =1**	
	A1	ReLU	
	Pool1	2×2, max pooling, stride =2	
	Conv2	5×5, **4**, **stride =1**	
	A2	ReLU	
	Pool2	2×2, max pooling, stride =2	
	Conv3	4×4, **5**, **stride =1**	
	A3	ReLU	
	Pool3	2×2, max pooling, stride =2	
Model-1 (b)	Conv1	5×5, **6**, **stride =1**	
	A1	ReLU	
	Pool1	2×2, max pooling, stride =2	
	Conv2	5×5, **8**, **stride =1**	
	A2	ReLU	
	Pool2	2×2, max pooling, stride =2	
	Conv3	4×4, **10**, **stride =1**	
	A3	ReLU	
	Pool3	2×2, max pooling, stride =2	

Model-1 (c)	Conv1	5×5**, 12, stride =1**	
	A1	ReLU	
	Pool1	2×2, max pooling, stride =2	
	Conv2	5×5, **16**, **stride =1**	
	A2	ReLU	
	Pool2	2×2, max pooling, stride =2	
	Conv3	4×4, **20**, **stride =1**	
	A3	ReLU	
	Pool3	2×2, max pooling, stride =2	
Model-2	Conv1	5×5, **3**, **stride =2**	
	A1	ReLU	
	Pool1	2×2, max pooling, stride =2	
	Conv2	5×5, **4**, **stride =2**	
	A2	ReLU	
	Pool2	2×2, max pooling, stride =2	
	Conv3	4×4, **5**, **stride =2**	
	A3	ReLU	
	Pool3	2×2, max pooling, stride =2	

Results

The comparison of the training and test accuracy and f-score rates obtained by ConNN classification based on mouth state information is shown in Table 3.

Table 3. The comparison of the training and test accuracy and f-score rates

Model	**ConNN**	**Epoch**	**YawDD**		**Nthu-DDD**		**KouBM-DFD**	
			Accuracy	***F-score***	***Accuracy***	***F-score***	***Accuracy***	***F-score***
Model-1 (a)	Training Accuracy	100	98.76%	96.59%	98.86%	96.51%	99.31%	96.94%
		500	99.27%	97.21%	99.09%	97.21%	99.73%	97.48%
		1000	99.70%	98.48%	99.30%	98.91%	99.79%	98.66%
	Test Accuracy	100	98.43%	95.77%	98.71%	95.31%	97.42%	95.52%
		500	99.20%	96.79%	98.78%	97.05%	98.89%	97.16%
		1000	**99.35%**	**98.01%**	**99.25%**	**98.32%**	**99.01%**	**97.95%**
Model-1 (b)	Training Accuracy	100	98.82%	96.02%	98.83%	96.48%	97.26%	94.24%
		500	98.92%	96.25%	98.89%	96.50%	97.84%	95.18%
		1000	98.90%	97.15%	98.87%	96.85%	97.87%	95.78%
	Test Accuracy	100	98.47%	95.65%	98.88%	95.21%	96.73%	94.02%
		500	98.60%	95.72%	98.69%	96.25%	96.82%	95.16%
		1000	**98.57%**	**96.95%**	**98.83%**	**96.42%**	**97.22%**	**95.24%**
Model-1 (c)	Training Accuracy	100	99.80%	97.48%	99.83%	96.83%	99.46%	95.98%
		500	99.89%	97.55%	99.91%	97.35%	99.82%	96.88%
		1000	99.91%	98.40%	99.32%	97.94%	99.86%	98.89%
	Test Accuracy	100	98.36%	96.70%	99.78%	96.28%	98.50%	94.88%
		500	98.71%	97.28%	99.21%	97.42%	98.83%	95.16%
		1000	**98.21%**	**97.35%**	**99.28%**	**98.10%**	**98.32%**	**96.56%**
Model-2	Training Accuracy	100	97.64%	95.28%	97.97%	94.81%	96.85%	94.74%
		500	97.77%	95.41%	97.88%	95.15%	97.39%	95.48%
		1000	97.80%	96.21%	97.94%	96.92%	97.73%	95.68%
	Test Accuracy	100	97.50%	94.81%	97.83%	94.82%	95.81%	94.26%
		500	97.57%	95.29%	97.81%	95.06%	96.59%	95.28%
		1000	**97.56%**	**96.58%**	**97.83%**	**96.56%**	**96.57%**	**95.86%**

The best accuracy rate was obtained in Model- (a). Accuracy and loss graphics on training and test data obtained in Model-1 (a) for the Yawdd dataset are shown in "Fig. 2". Accuracy and loss graphics on training and test data obtained in Model-1 (a) for the NthuDD dataset are shown in "Fig 3". Accuracy and loss graphics on training and test data obtained in Model-1 (a) for the KouBm-DFD dataset are shown in "Fig. 4".

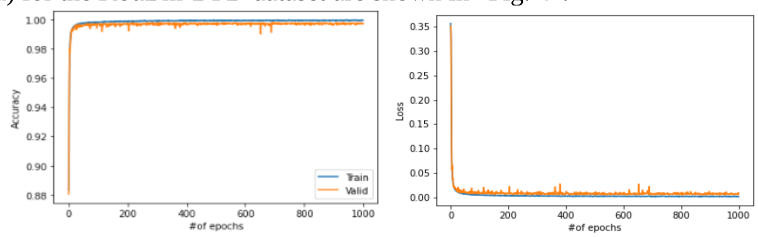


Fig. 2. Accuracy and loss graphics on training and test data in the YawDD Dataset

During the training phase, it was observed that the training and test error tended to decrease rapidly until the number of cycles reached approximately 30. It was observed that this rapid decline of the test error slowed down when the number of cycles reached approximately 45, but the training error slowed down when the number of cycles reached approximately 55 cycles. The model has been observed up to 1000 cycles in the graph and by looking at the error function, it is concluded that even if the oscillations continue, it remains the same and does not change, so the learning of the model is over.

Similarly, it has been observed that both training and test accuracy tend to increase rapidly at the same time until the number of cycles reaches approximately 30 during the training phase. It was observed that this rapid increase in test accuracy slowed down when the number of cycles reached approximately 45, but the training accuracy slowed down when the number of cycles reached approximately 55. The model has been observed up to 1000 cycles in the graph and by looking at the error function, it is concluded that even if the oscillations continue, it remains the same and does not change, so the learning of the model is over.

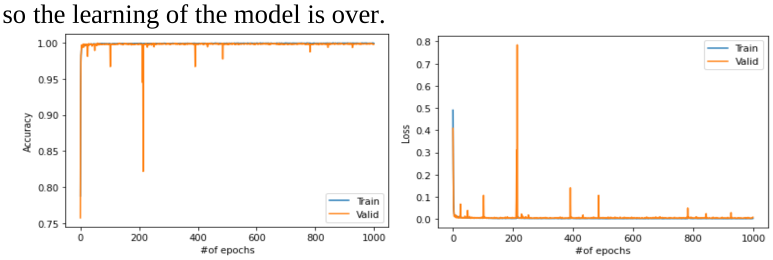


Fig. 2. Accuracy and loss graphics on training and test data in the Nthu_DDD Dataset

During the training phase, it was observed that the training and test error tended to decrease rapidly until the number of cycles reached approximately 15. It was observed that this rapid decline of the test error slowed down when the number of cycles reached approximately 23, but the training error slowed down when the number of cycles reached approximately 30 cycles. The model has been observed up to 1000 cycles in the graph and by looking at the error function, it is concluded that even if the oscillations continue, it remains the same and does not change, so the learning of the model is over.

Similarly, it has been observed that both training and test accuracy tend to increase rapidly at the same time until the number of cycles reaches approximately 15 during the training phase. It was observed that this rapid increase in test accuracy slowed down when the number of cycles reached approximately 23, but the training accuracy slowed down when the number of cycles reached approximately 30. The model has been observed up to 1000 cycles in the graph and by looking at the error function, it is concluded that even if the oscillations continue, it remains the same and does not change, so the learning of the model is over.

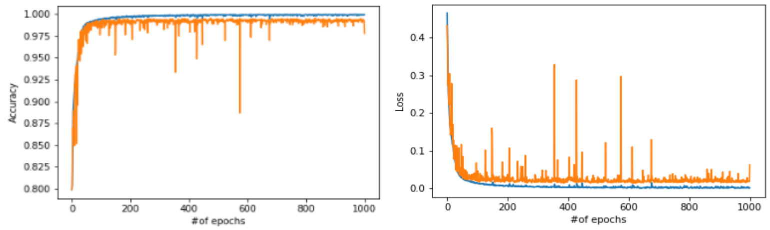


Fig. 3. Accuracy and loss graphics on training and test data in the KouBm-DFD Dataset

During the training phase, it was observed that the training and test error tended to decrease rapidly until the number of cycles reached approximately 70. It was observed that this rapid decline of the test error slowed down when the number of cycles reached approximately 110, but the training error slowed down when the number of cycles reached approximately 200 cycles. The model has been observed up to 1000 cycles in the graph and by looking at the error function, it is concluded that even if the oscillations continue, it remains the same and does not change, so the learning of the model is over.

Similarly, it has been observed that both training and test accuracy tend to increase rapidly at the same time until the number of cycles reaches approximately 70 during the training phase. It was observed that this rapid increase in test accuracy slowed down when the number of cycles reached approximately 110, but the training accuracy slowed down when the number of cycles reached approximately 200. The model has been observed up to 1000 cycles in the graph and by looking at the error function, it is concluded that even if the oscillations continue, it remains the same and does not change, so the learning of the model is over.

Comparison of the success rate achieved with other approaches in the literature is shown in Table 4. Accuracy rates were found to be higher and more successful than other approaches.

Table 4. The comparison of the accuracy rates

	Model	**Dataset**	**Accuracy**
Zhang et al., 2015 [16]	CNN	Their own data set, YawDD	89%
Yan et al., 2015 [17]	CNN	Their own data set	98.22%
Zhang et al., 2017 [11]	CNN	ImageNet, State Farm Distracted Driver Detection, YawDD	87%
Gu et al., 2018 [19]	MTCNN	EMD, ZJU, CEW	98.05%
Zhao et al., 2020 [20]	EM-CNN	Biteda	93.62%
Proposed model	ConNN Model-1 (a)	YawDD, Nthu-DDD, KouBM-DFD	99.35% 99.25% 99.01%

IV. Conclusion

Driver fatigue detection is one of the important and difficult one of the most important and difficult problems to solve in the literature because this problem depends on human beings and every person's reaction to fatigue is variable. The aim of this study is to design a driver fatigue detection system compatible with real-time systems, which can be performed with high accuracy the test phase like training phase, for systems that require immediate intervention to the driver or not. In this context, the ConNN model has been proposed.

In the study, the driver's yawning determination and the number of yawning were used to determine driver fatigue. When the results obtained are evaluated, driver fatigue was detected for YawDD, Nthu-DDD and KouBM-DFD data sets with accuracy rates of 99.35%, 99.25%, and 99.01%, respectively.

References

[1]. W. A. Cobb, “Recommendations for the practice of clinical neurophysiology,” Elsevier, Amsterdam,1983.
[2]. X. Yu, “Real-time nonintrusive detection of driver drowsiness,” Technical Report for University of Minnesota: Minneapolis, MN, USA, 2009.
[3]. R. Feng, G. Zhang, B. Cheng, “An on-board system for detecting driver drowsiness based on multi-sensor data fusion using Dempster-Shafer theory,” Proc. in Networking ICNSC, Okayama, Japan, 2009.

[4].
[5]. M. Ingre, T. Åkerstedt, B. Peters, A. Anund, and G. Kecklund, "Subjective sleepiness, simulated driving performance and blink duration: Examining individual differences," J. Sleep Res., 2006, 15(1), 47–53.
[6]. C. Lavergne, P. De Lepine, P. Artaud, S. Planque, A. Domont, C. Tarriere, C. Arsonneau, X. Yu, A. Nauwink, C. Laurgeau, J.M. Alloua,R. Y. Bourdet, J.M. Noyer, S. Ribouchon, and C. Confer, "Results of the feasibility study of a system for warning of drowsiness at the steeringwheel based on analysis of driver eyelid movement," Proc. 15th Int. Tech. Conf. Enhanced Safety Vehicles, 1996, Vol. 1,.
[7]. L. Hartley, T. Horberry, N. Mabbott, and G P. Krueger, "Review of fatigue detection and prediction Technologies", Technical report, NationalRoad Transport Commission, 2000.
[8]. B. K. Savas, Y. Becerikli, "Development of driver fatigue detection system by using video images," Akıllı sistemlerde Yenilikler Ve Uygulamalar Konferansı, Alanya/Turkey, 2017.
[9]. B. K. Savas, Y. Becerikli, ``Real time driver fatigue detection based on SVM algorithm," Proc. 6th Int. Conf. Control Eng. Inf. Technol. (CEIT), 25-27 Oct., İstanbul/Turkey, 2018.
[10]. P. Viola and M. J. Jones, "Robust real-time object detection", ICCV Workshop on Statistical and Computation Theories of Vision, 2004, 57(2), 137-154.
[11]. Freund, Y., Schapire, R. E., "Experiments with a New Boosting Algorithm", Lecture Notes in Computer Science (Lecture Notes in Artificial Intelligence), Springer, 1995, 904, 148-156,.
[12]. F. Zhang J. Su L. Geng, Z. Xiao "Driver fatigue detection based on eye state recognition," Proc. - 2017 Int. Conf. Mach. Vis. Inf. Technol. C., Singapore/Singapore, 2017.
[13]. D. Scherer, A. Müller, and S. Behnke, "Evaluation of Pooling Operations in Convolutional Architectures for Object Recognition," In Proc. International conference on artificial neural networks, Springer, Heidelberg/ Berlin, 2010.
[14]. B. G. Pratama, I. Ardiyanto, T. B. Adji, "A review on driver drowsiness based on image, bio-signal, and driver behavior," IEEE Int. Conf. on Science and Technology-Computer, Yogyakarta/Indonesia, 2017.
[15]. H. B. Kang, "Various approaches for driver and driving behavior monitoring: A review," Proc. IEEE Int. Conf. Comput. Vis., Aralık, 2013.
[16]. Q. Ji, Z. Zhu, P. Lan., "Real-time Nonintrusive monitoring and prediction of driver fatigue", IEEE Transactions on Vehicular Technology, 2004, 53(4), 1052–1068.
[17]. W. Zhang, Y. L. Murphey, T. U. Wang, Q. Xu, "Driver yawning detection based on deep convolutional neural learning and robust nose tracking," IEEE International Joint Conference on Neural Networks (IJCNN),12-17 Jul., Killarney/Ireland, 2015.
[18]. Lai H., Xiao S., Cui Z., Pan Y., Xu C., Yan S., Deep cascaded regression for face alignment, arXiv preprint arXiv:1510.09083, 2015, 1.
[19]. C. Yan, H. Jiang, B. Zhang, F. Coenen, "Recognizing driver inattention by convolutional neural networks," International Congress on Image and Signal Processing (CISP), 14-16 Oct., Shenyang/China, 2015.
[20]. W.H. Gu, Y. Zhu, X.D. Chen, L.F. He, B.B. Zheng, "Hierarchical CNN-based real-time fatigue detection system by visual-based technologies using MSP model," IET Image Processing, 2018, 12(12), 2319-2329.
[21]. Zhao, Z., Zhou, N., Zhang, L., Yan, H., Xu, Y., & Zhang, Z. Driver Fatigue Detection Based on Convolutional Neural Networks Using EM-CNN. Computational Intelligence and Neuroscience, 2020.
[22]. VentureBeat, "Autonomous cars could end drowsy driving, but don't hold your breath," [Online]. Available: https://venturebeat.com/2017/11/19/autonomous-cars-couldend-drowsy-driving-but-dont-hold-your-breath/. [Accessed: 14-Jan-2020].
[23]. Bosch, "CES® 2017: Bosch is showing these smart solutions in Las Vegas - Bosch Media Service." [Online]. Available: http://www.bosch-presse.de/pressportal/de/en/ces-2017-bosch-is-showing-these-smart-solutions-in-las-vegas-81792.html. [Accessed: 16-Nov-2020].
[24]. nvidia, "NVIDIA DRIVE IX Self-Driving Vehicles | NVIDIA Automotive." [Online]. Available: https://www.nvidia.com/en-us/self-driving-cars/drive-ix/. [Accessed: 16-Nov-2020].
[25]. B. K. Savas, Y. Becerikli, "Real Time Driver Fatigue Detection System Based on Multi-Task ConNN," IEEE Access, 2020, 8, 12491-12498.
[26]. Junaedi S., Akbar H., Driver Drowsiness Detection Based on Face Feature and PERCLOS, International Conference on Computation in Science and Engineering, Bandung, Endonezya, 10-12 July 2017.
[27]. S. Abathi, M. Omidyeganeh, S. Shirmohammadi, B. Hariri, "YawDD: a yawning detection dataset," Proc. 5th ACM Multimedia Syst. Conf, Singapore, 2014.
[28]. C. Weng, Y. Lai, S. Lai, "Driver Drowsiness Detection via a Hierarchical Temporal Deep Belief Network", In Asian Conference on Computer Vision Workshop on Driver Drowsiness Detection from Video, Taipei,/Taiwan, 2016.
[29]. B. K. Savas, Y. Becerikli, "Derin öğrenme tabanlı sürücü yorgunluğu sezme ve tahmin sistemi geliştirilmesi," Doctoral Thesis, Kocaeli University, Turkey, 2021.

Burcu Kır Savaş, et. al. "A Deep Learning Approach to Driver Fatigue Detection via Mouth State Analyses and Yawning Detection." *IOSR Journal of Computer Engineering (IOSR-JCE)*, 23(3), 2021, pp. 24-30.