Eyes Detector Approach for Driving Monitoring System for Occlusion Faces without using Facial Landmarks

1st Myriam Vaca-Recalde
Tecnalia
Basque Research and Technology Alliance (BRTA), Edif 700
Derio, Spain
myriam.vaca@tecnalia.com, 0000-0002-7762-524X

2nd Pedro López-García
Tecnalia
Basque Research and Technology Alliance (BRTA), Edif 700
Derio, Spain
pedro.lopez@tecnalia.com, 0000-0002-0515-4928

3rd Javier Echanobe
Department of Electricity and Electronics
University of the Basque Country
Leioa, Spain
franciscojavier.echanove@ehu.eus, 0000-0002-1064-2555

4th Joshué Pérez
Tecnalia
Basque Research and Technology Alliance (BRTA), Edif 700
Derio, Spain
joshue.perez@tecnalia.com, 0000-0001-8328-9978

***Abstract*—The current health situation with the use of masks complicates the analysis of gaze and head direction in driver monitoring systems based on facial detection since landmarks are not working properly. Due to this issue, the need to solve occlusion problems using an alternative method to the current ones has increased. On the other hand, the deployment of these systems inside the vehicles must be carried out in the least intrusive way possible for the driver. This article presents an approach for driver distraction analysis based on the driver's eyes without using landmarks applying Deep Learning methods, and the study of different parameters such as detection speed for the deployment of the best accuracy-speed method in an embedded platform. Different state-of-the-art and open source neural networks have been used and tuned to address our current problem. On the other hand, as is well known, training these models requires an enormous amount of data. In the case of gaze, there are very few data sets dedicated specifically to it. UnityEyes software has been used to create the training and test datasets for the system since it creates the necessary amount of data needed by the models easily.**

***Index Terms*—Driving Monitoring System (DMS), Artificial Intelligence, Advanced Driver Assistance System (ADAS), Deep Learning.**

I. Introduction

The concern to reduce traffic accidents has led to the development of Advanced Driver Assistance Systems (ADAS) among other solutions. Specifically, the importance of Driving Monitoring Systems (DMS) based on the study of driver distractions has increased, since they are one of the main reasons for road accidents [1] [2]. The driver can be distracted by audio, mechanical, cognitive, or visual distractions [3]. Focusing on visual distractions, these refer to situations where the driver takes his/her eyes off the road and may involve a momentary head rotation. In general, the treatment of these types of distractions depend on detecting and tracking facial landmarks on the user [4]. Usually this detection is carried out through the use of cameras and image analysis to obtain facial features that allow locating head, eyes, mouth, etc., and define if the driver is distracted or not. However, occlusions have always been one of the challenges in this type of method [5] [6].

Nowadays, with the use of the mask due to the health emergency caused by COVID-19, studies related to the subject have emerged and have led to masked faces datasets for different uses such as facial detection [7], analysis of the correct position mask [8], etc. With the use of the mask, the systems have more difficulties to detect facial landmarks properly. For DMS, this is a big challenge since eyes are the only viable feature in those cases.

On the other hand, the advancement of technology has allowed an increase in computational power with modern GPUs and their parallelization, providing improvements in the research of Deep Learning applying on DMS systems, allowing the application of advanced network architectures [9]. However, detecting masked faces remains a challenging task for many existing models due to masked faces could have different orientations, degrees of occlusion, or different types of masks, making the poor accuracy of these detections a problem to be solved. In addition, as far as authors concerns, there is not a large enough dataset to do a correct exploration of the key attributes and to use this data to identify them, and train and test the models in a proper way. So, with insufficient training and testing data as well as incomplete and inaccurate features, masked face detection is a widely challenging task.

Our contribution in this work relies on creating a system capable of offering real-time information about the driver distraction based on the gaze with the main occlusion of the mask, and without depending on the need to obtain the facial features to detect the components of the face such as the mouth, chin, etc. For that, Deep Learning methods, especially open source pretrained neural networks, have been applied to achieve the best accuracy-speed method. For the preliminary dataset, UnityEyes software has been used to create both training and the initial testing dataset of the system since it creates the necessary amount of data easily. The proposed system is part of the development of a complete DMS that attempts to detect driver distraction at all times regardless of whether or not their face is occluded, in real time and with a low computational load.

This article is divided as follows. Section II presents the State of the Art of Facial Recognition systems, describing their evolution and how the problem of occlusions has been dealt with. Section III contains a description of the system focusing on the proposed approach to detect the eyes without relying on facial features. Section IV describes the details of the experimental tests and results focused on comparing the effectiveness of different neural networks for this application. Finally, the article ends by presenting the conclusions and future works in Section V.

II. Related Work

Facial Recognition (FR) is one of the most important non-intrusive biometric techniques for authenticating people and is mainly based on computer vision techniques in application fields such as security, military, surveillance, etc.

Deep Learning (DL) has reshaped the FR research landscape by improving the performance of these systems and their applications in real world scenarios. DL applies multiple layers of processing to learn representations of data with multiple levels of feature extraction. Since the appearance of DeepFace [10] and DeepID [11] in 2014, the advancement and improvement of these processes have increased dramatically. In addition, access to a large number of facial images, including public datasets [12] [13], has helped to improve the performance of facial recognition. Although it must be taken into account that there are still the problems of variable lighting, low resolution, different facial expressions, and occlusion. Generally, images stored in databases are free of these defects, which affects training tests with real-world testing. In [14] and in [15] a complete vision of recent developments is presented, describing different algorithms, databases, protocols, and scenarios.

From the point of view of driving monitoring, FR is one of the most widely used and fastest non-intrusive techniques to know the driver's condition by detecting distraction [16]. In real driving situations, obtaining information in real time is critical to take any action. [17] demonstrated that for every 25% increase in total glance duration, reaction time is increased by 0.39 seconds and standard deviation of lane position is increased by 0.06 meters in real time diving simulator tests. Research into driver distraction detection has increased and achieved great results with the application of DL. A summary of modern neural-network-based facial landmark detection algorithms is described in [18]. Finally, a review of the role of computer vision technology applied to the development of monitoring systems to detect distraction is described in [19]. However, as mentioned in [5], occlusions are a challenging problem in FR systems due to the lack of information caused by the error in the localization of facial features and the type of the occlusions. Moreover, specific datasets of occluded faces such as [8] or [20] have also been created, to help in the development of research in this field.

A survey describing how existing face recognition methods cope with the occlusion problem is presented in [21]. It should be noted that almost none of the methods presented in this survey have been tested in real-time ADAS applications such as driver monitoring, although in applications like that it is a recurring problem. In the literature of DMS, most of the procedures have sought to develop systems, that are sufficiently robust to be able to detect distraction or drowsiness despite occlusions without facing them directly such as [22] or [23] where the authors tested that their system is robust enough to occlusion of the eyes. The approach presented in this work focuses on directly solving the occlusion problem of the middle of the face giving a robust and reliable solution without taking into account the facial features.

III. System Description

This section is divided in the following way. First, Section III-A describes the whole DMS system and its current state so far. Section III-B contains the detailed definition of the system proposed in this article, which focuses on eyes detection. It should be emphasized that in this paper the authors want to show the progress of the work carried out during the research and development of this system as well as its virtues and defects.

A. *System flowchart*

Figure 1 shows the system as it is developed at the moment this article is presented. A camera captures the driver's face inside the car and processes the image to detect whether or not there are occlusions that prevent obtaining the necessary characteristics for facial analysis. The camera is a Basler camera acA1920-40uc, that delivers 41 frames per second at 2.3 MP resolution, with a 12 mm focal length lens. To detect the occlusion, the system is trained with the database described in [24], which was created because there are no available large datasets of masked face images that allow working with mask occlusions not necessarily well placed.

Once the face is detected, it is classified into ”occluded” or ”not occluded”. If it is not occluded, it will use the system developed and described in [25], taking into account landmarks, and indicating whether the driver is distracted or not, using CNN models. In case of the face is occluded or it is not detecting any of the necessary points to obtain the face features, for example when the head moves to its extreme positions, the image will be processed by the system proposed using the presented approach described in detail in the next section.

The output after the eyes detection would be the gaze angle and the head position. The data is created using Unity Eyes software [26]. The information the software provides will be used to help with the determination of these characteristics in future works. This paper is focused on the detection of the eyes without using landmarks applying different open source deep neural networks. Finally, the outputs obtained by one of the two systems are the inputs of the Distraction Classifier.

B. Eyes detection module

As it is already known, deep neural networks need a huge amount of data to be trained successfully. Due to this, there are certain difficulties, being one of the most important ones the lack of open datasets for specific problems, as it is the case we are dealing with. The system pretends to detect eyes without taking into account landmarks or other significant parts of the user's face. Due to the aforementioned problem with datasets, Unity Eyes software [26] has been used since it allows us to randomly create as many eye images as necessary without needing to record real people or cropping the eye area from wild images of other datasets and label them. An example of the images provided by Unity Eyes can be shown in Figure 2.

Once the dataset is created, the position of the eye within the image is extracted. Subsequently, the eye zone feed the neural networks used in the experimentation, performing a regression to obtain the associated bounding box. For this article, it has been necessary a total of 2000 images to train the networks. Once they are trained, the initial testing is carried out on images of the same type. It has been used 400 images to create the test set. Although, later the best networks so far will be performed on real images for other State of the Art datasets.

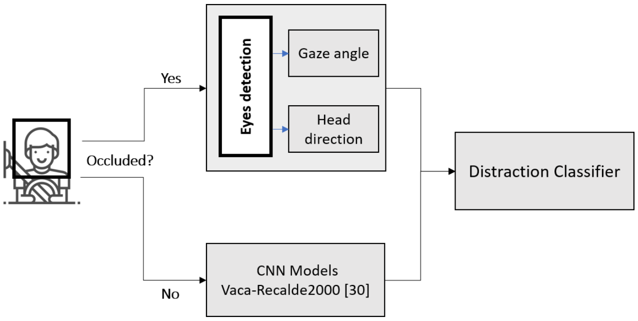

Fig. 1. System Flowchart.

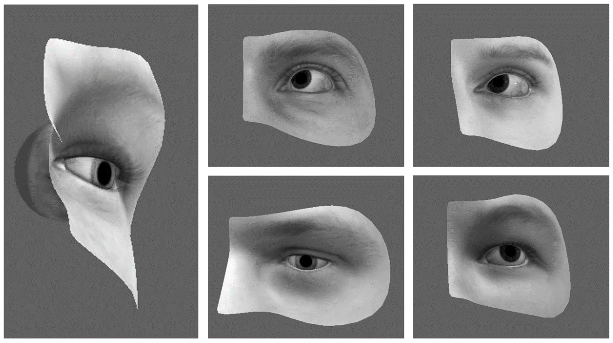

Fig. 2. Examples of the Unity Eyes images

Figure 3 shows the final flowchart of the eye detector: the input to the system will be a face image and the output will be the coordinates of the bounding box that contains the eye.

IV. Experimentation and Results

This section contains the experimentation carried out in this development divided in three parts. Section IV-A contains the definition and configuration used in the different deep neural networks. Section IV-B shows the results of the experimentation. Finally, its application in real-world images is shown in Section IV-C.

A. Models used

One of the purposes of this work is to make a system that is replicable by other researchers easily and comfortably. For this, the Tensorflow framework and the models offered by it have been used [1]. In this work, four models were compared to decide which would be the most optimal to use in this application. Model definitions are mentioned below.

- EfficientDet D0 [27] is part of a family of models developed by Google Brain and built on top of EfficientNet.
- Faster R-CNN ResNet50 [28] evolved from R-CNN and Fast R-CNN. It is principally used for object detection. It is developed around different backbones. In this case, ResNet50 [29] is used.
- Faster R-CNN ResNet152 [30]. Same as previous network, using ResNet152 backbone.
- SSD Mobilenet FPNLite [31]. SSD Mobilenet uses a subnetwork called Feature Pyramid Network which outputs feature maps of different resolutions.

Configuration of the networks has been set to the default ones. Only training steps have been modified. This experimentation is carried out in an AMD Ryzen 9 3950X 16-Core with 32 GB RAM.

[1] https://github.com/tensorflow/models/blob/master/research/object_detection/g3doc/tf2_detection_zoo.md

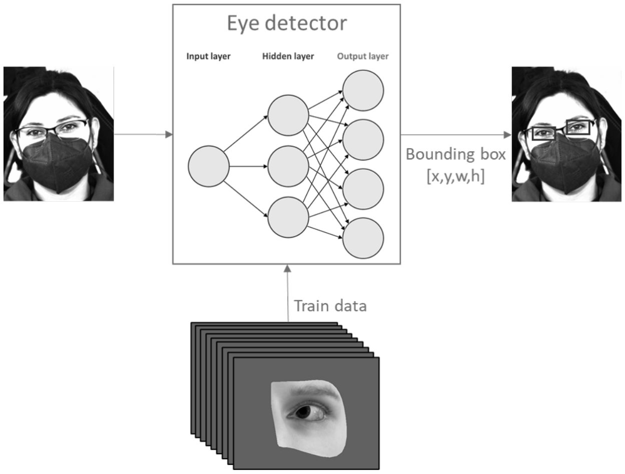

Fig. 3. Diagram of the proposed system

B. Results

This section shows the results achieved from the experiments carried out to evaluate the presented system. Precision and recall metrics are taking into account to check the performance of the different used networks. The networks have been trained for $3,000$, $5,000$, and $10,000$ steps in this first approach. Table I contains the values for the different metrics. Bold values represent the best values so far for each number of steps.

In general, FR-CNN Resnet152 obtains both the best precision and recall so far in all cases. At $5,000$ and $10,000$ steps, FR-CNN Resnet50 obtains similar results to the previous network in recall and precision respectively. The EfficientDet D0 network has a noticeable improvement as the steps increase, however it does not reach the values as high as the ResNets. Finally, the FPNLite network has good values and is close to ResNets at $10,000$ steps.

The improvement of the models between steps ($3,000$ to $5,000$ and $3,000$ to $10,000$) and the Detection Time (DT) in seconds is exposed in Table II. This last characteristic is important since the model is going to be used in a real-time system. The best improvement is made by EfficientDet D0 from $3,000$ to $10,000$ steps. However, it is FPNLite the model that improves the most if the DT is taking into account. Since this experimentation looks for the model that obtains the best performance, also taking into account the inference time for new images, the model with the best precision-time measurement is **FPNLite** trained for $10,000$ steps. Time-precision is defined as the division of the precision obtained by the networks between the detection time.

TABLE I
NETWORKS RESULTS WITH DIFFERENT STEPS

Steps	Model name	Precision	Recall
3000	EfficientDet D0	.68	.72
	FR-CNN ResNet50	.88	.91
	FR-CNN ResNet152	**.89**	**.92**
	FPNLite	.82	.85
5000	EfficientDet D0	.73	.76
	FR-CNN ResNet50	.91	**.94**
	FR-CNN ResNet152	**.92**	**.94**
	FPNLite	.87	.89
10000	EfficientDet D0	.78	.82
	FR-CNN ResNet50	**.92**	**.94**
	FR-CNN ResNet152	.91	**.94**
	FPNLite	.89	.92

TABLE II
IMPROVEMENT OF THE MODELS AND DETECTION TIME

Model name	3k-5k	3k-10k	DT(s)	Precision vs Time
EfficientDet D0	.05	0.1	1.07	.73
FR-CNN ResNet50	.03	.03	1.23	.75
FR-CNN ResNet152	.03	.03	1.81	.50
FPNLite	.05	.07	**0.62**	**1.43**

C. Real World application

This system is intended to work as a module of a DMS as explained in Figure 1. Therefore, although the proposed system has been trained with synthetic values from UnityEyes, its final function is to be used with real images as shown in Figure 3. However, to our knowledge, none of the open datasets include a bounding box detection of the eyes of real people images and a way to be compared. Hence, the authors leave the comparison with real images for a later version of the system, focusing the comparison only on the results expressed in the previous section.

Some examples of the best results networks in real images, both with and without occlusion are shown in Figure 4 . As can be seen in the images, the detection of the eye is correct in most of the cases, and the corresponding bounding boxes are obtained despite the occlusions and without them. It is important to emphasize that the system that has obtained the best results in ”Precision vs Time” metric (FPN-Lite) is not the best in real-time images but the ResNet results in the images are very promising. These conclusions will be taken into account and studied in future works.

V. CONCLUSIONS AND FUTURE WORKS

An early version of a driver’s eyes detection system has been introduced that does not depend on the different characteristics of the face, thus being appropriate for detecting masked faces. The system is based on open source networks which are fine-tuned for our particular problem. For this tuning, a dataset of eyes images was created by using the tool Unity Eyes which can create many random images as needed by the user and different characteristics of the eyes.

For the experiments, four different networks have been used and obtained results show rather good results both in recognition and in processing speed. In particular, the neural network knows as *SSD Mobilenet FPNLite* provides the best trade-off between classification rate and detection time. Thus, it can obtain around 0.9 both in Precision and Recall and perform the process in just 0.62 seconds. However, this network does not obtain good results with real world data, which is the final intention of the system.

In the light of the results, the system applies to a real-time driving monitoring system aimed to detect driver distractions. The system presented here is part of a larger system - which is being developed by our group - in which the ultimate objective is to detect driver distractions, both if the face is masked or it is fully visible.

As future works, the number of networks that have been experimented with as well as exploration of the appropriate size of the training data set will be tested. The results obtained in this work will be used in the whole occlusion system and gaze angle and head direction will be determined without taking into account landmarks. Hyperparameter tuning would be also considered as well as the creation of open datasets related to occlusion problem.

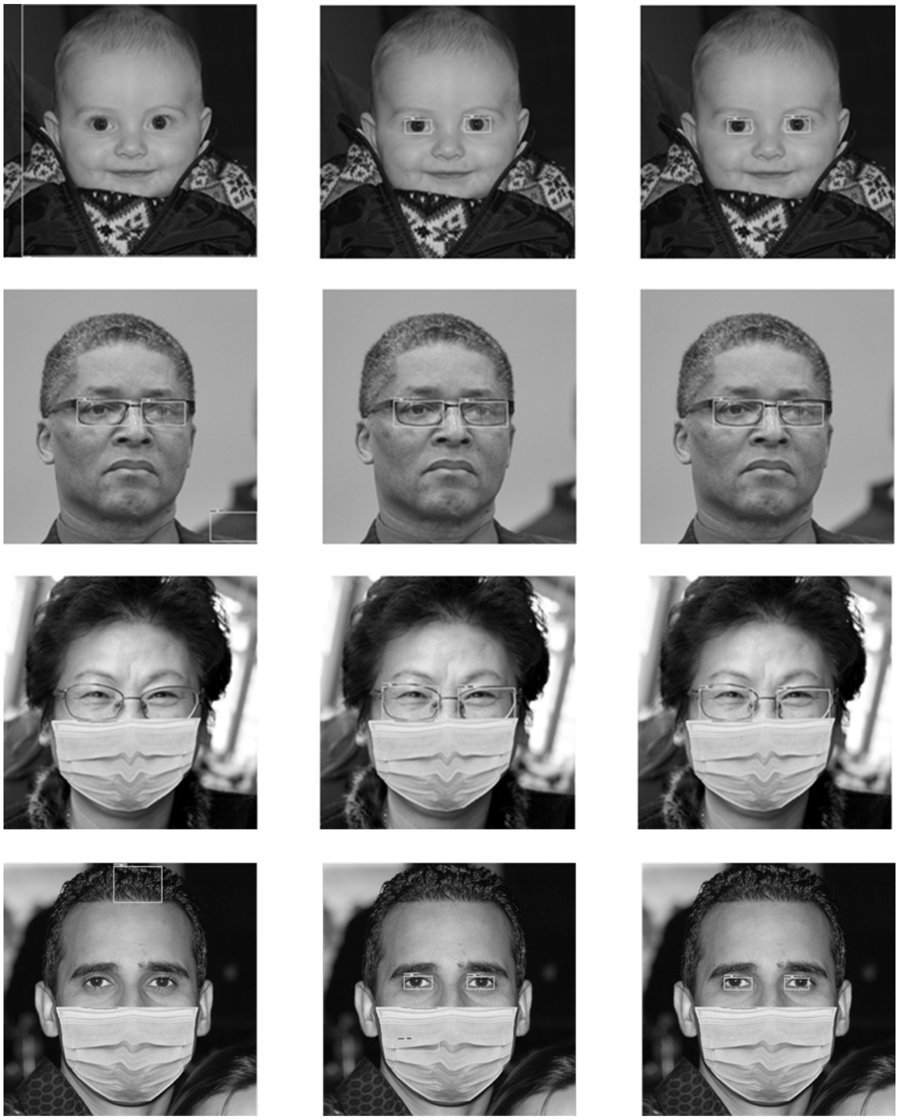

Fig. 4. Examples of the results in real images for FPNLite (left column), FR-CNN ResNet50 (center) and FR-CNN ResNet152 (right column)

ACKNOWLEDGMENT

This work has been supported by HADRIAN project, which has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 875597. Also authors want to thank to the Spanish Ministry of Economy, Industry and Competitiveness TEC2016- 77618-R (AEI/FEDER, UE), and the Basque Country University UPV/EHU under Grant GIU18/22 for their support in the development of this work.

REFERENCES

[1] NHTSA, "Research Note Distracted Driving in Fatal Crashes, 2017," *NHTSA's National Center for Statistics and Analysis*, vol. DOT HS 812, no. April, pp. 1–8, 2019.

[2] G. Fountas, S. Sonduru Pantangi, S. Shahriar Ahmed, U. Eker, and P. C. Anastasopoulos, "FINAL REPORT Factors affecting perceived and observed aggressive driving behavior: An empirical analysis of driver fatigue, and distracted driving," Tech. Rep.

[3] F. Prat, M. E. Gras, M. Planes, S. Font-Mayolas, and M. J. Sullman, "Driving distractions: An insight gained from roadside interviews on their prevalence and factors associated with driver distraction," *Transportation Research Part F: Traffic Psychology and Behaviour*, vol. 45, pp. 194–207, feb 2017.

[4] G. Sikander and S. Anwar, "Driver fatigue detection systems: A review," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 6, pp. 2339–2352, 2019.

[5] H. K. Ekenel and R. Stiefelhagen, "Why is facial occlusion a challenging problem?" in *Advances in Biometrics*, M. Tistarelli and M. S. Nixon, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 299–308.

[6] M. O. Oloyede, G. P. Hancke, and N. Kapileswar, "Evaluating the effect of occlusion in face recognition systems," in *2017 IEEE AFRICON*, 2017, pp. 1547–1551.

[7] Q. Ye, "Masked face detection via a novel framework," in *Proceedings of the 2018 International Conference on Mechanical, Electronic, Control and Automation Engineering (MECAE 2018)*. Atlantis Press, 2018/03, pp. 238–243. [Online]. Available: https://doi.org/10.2991/mecae-18.2018.137

[8] A. Cabani, K. Hammoudi, H. Benhabiles, and M. Melkemi, "Maskedface-net – a dataset of correctly/incorrectly masked face images in the context of covid-19," *Smart*

Health, vol. 19, p. 100144, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2352648320300362

[9] M. Ngxande, J.-R. Tapamo, and M. Burke, "Driver drowsiness detection using behavioral measures and machine learning techniques: A review of state-of-art techniques," in *2017 Pattern Recognition Association of South Africa and Robotics and Mechatronics (PRASA-RobMech)*, 2017, pp. 156–161.

[10] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, "Deepface: Closing the gap to human-level performance in face verification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014.

[11] X. W. Y. Sun, Y. Chen and X. Tang, "Deep learning face representation by joint identification-verification," in *Advances in neural information processing systems*, 2014, p. 1988–1996.

[12] N. Gourier, D. Hall, and J. Crowley, "Estimating face orientation from robust detection of salient facial structures," *FG Net Workshop on Visual Observation of Deictic Gestures (POINTING)*, pp. 17–25, 2004. [Online]. Available: http://www.homepages.inf.ed.ac.uk/rbf/CAVIAR/PAPERS/Pointing04-Gourier.pdf

[13] B. Luo, J. Shen, Y. Wang, and M. Pantic, "The iBUG Eye Segmentation Dataset," in *2018 Imperial College Computing Student Workshop (ICCSW 2018)*, ser. OpenAccess Series in Informatics (OASIcs), E. Pirovano and E. Graversen, Eds., vol. 66. Dagstuhl, Germany: Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2019, pp. 7:1–7:9.

[14] M. Wang and W. Deng, "Deep face recognition: A survey," *CoRR*, vol. abs/1804.06655, 2018.

[15] I. Masi, Y. Wu, T. Hassner, and P. Natarajan, "Deep face recognition: A survey," in *2018 31st SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*, 2018, pp. 471–478.

[16] M. hoseyn Sigari, M. reza Pourshahabi, M. Soryani, and M. Fathy, "A review on driver face monitoring systems for fatigue and distraction detection," pp. 73–100, 2014.

[17] H. Zhang, M. R. H. Smith, and G. J. Witt, "Identification of real-time diagnostic measures of visual distraction with an automatic eye-tracking system," *Human Factors*, vol. 48, no. 4, pp. 805–821, 2006.

[18] K. Khabarlak and L. Koriashkina, "Fast facial landmark detection and applications: A survey," 2021.

[19] A. Fernández, R. Usamentiaga, J. L. Carús, and R. Casado, "Driver distraction using visual-based sensors and algorithms," *Sensors*, vol. 16, no. 11, 2016.

[20] S. Lin, L. Cai, X. Lin, and R. Ji, "Masked face detection via a modified lenet," *Neurocomputing*, vol. 218, pp. 197–202, 2016.

[21] D. Zeng, R. Veldhuis, and L. Spreeuwers, "A survey of face recognition techniques under occlusion," 2020.

[22] K. Yuen, S. Martin, and M. M. Trivedi, "Looking at faces in a vehicle: A deep cnn based approach and evaluation," in *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, 2016, pp. 649–654.

[23] T. D'Orazio, M. Leo, C. Guaragnella, and A. Distante, "A visual approach for driver inattention detection," *Pattern Recognition*, vol. 40, no. 8, pp. 2341–2355, 2007, part Special Issue on Visual Information Processing.

[24] A. Cabani, K. Hammoudi, H. Benhabiles, and M. Melkemi, "Maskedface-net – a dataset of correctly/incorrectly masked face images in the context of covid-19," *Smart Health*, vol. 19, 2021.

[25] M. E. Vaca-Recalde, J. Pérez, and J. Echanobe, "Driver Monitoring System Based on CNN Models: An Approach for Attention Level Detection," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 12490 LNCS. Springer Science and Business Media Deutschland GmbH, nov 2020, pp. 575–583.

[26] E. Wood, T. Baltrušaitis, L.-P. Morency, P. Robinson, and A. Bulling, "Learning an appearance-based gaze estimator from one million synthesised images," in *Proceedings of the Ninth Biennial ACM Symposium on Eye Tracking Research Applications*, 2016, pp. 131–138.

[27] M. Tan, R. Pang, and Q. V. Le, "Efficientdet: Scalable and efficient object detection," 2020.

[28] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," 2016.

[29] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2015.

[30] B. Cheng, Y. Wei, H. Shi, R. Feris, J. Xiong, and T. Huang, "Revisiting rcnn: On awakening the classification power of faster rcnn," 2018.

[31] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," 2017.