Turkish Journal of Computer and Mathematics Education Vol.12 No.9 (2021), 1892 - 1898

Research Article

Driver-Drowsiness Detection System

S.Karthikeyan[a], R.Subha[b], G.Elakkiya[c], R.Kowshika[d], and S.Dhanvarshni[e]
a,b,c,d,e
Department of Information Technology
M.Kumarasamy College of Engineering, Thalavapalayam, Karur-639113
karthikeyans.it@mkce.ac.in, subharamesh2000@gmail.com, elakkiyanavya@gmail.com, dhanvarshni13@gmail.com, kowshikaravi99@gmail.com

Article History: Received: 10 January 2021; Revised: 12 February 2021; Accepted: 27 March 2021; Published online: 20 April 2021

Abstract: Due to the drowsiness of drivers, car accidents kill thousands of people worldwide every year.This fact clearly illustrates the need for a sleep sensor application to help prevent such accidents and ultimately save lives. In this work, we propose a novel intensive learning method based on neutral neural networks (CNN) to deal with this problem. In this project we aim to develop a prototype drowsiness detection system. The system works by monitoring the driver's eyes and ringing the alarm while it is drying.

The system is a real-time control system that is not intrusive. The priority is to improve driver safety without intrusion. In this project, the driver's eyelid is detected. When a driver's eyes are closed for an extended period of time, the driver is considered indifferent, and an alarm rings. The Haar Cascade library is used to detect facial features, and programming is performed in OpenCV.

1. Introduction

Driver shortcoming could be a colossal calculate in an colossal number of vehicle incidents. Continuous estimates indicate that the number of passes associated with unfortunate shortcomings is 1,200 and the number of casualties is 76,000 each year.

The development of technologies to detect or prevent drowsiness in the driver's seat is an important test in the field of accident prevention systems. In light of the dangers that laziness poses in public, strategies for counteracting its effects should be created.

The goal of this project is to create a model language recognition framework. The focus will be on developing a framework that can continuously monitor whether the driver's eyes are open or closed. It is widely recognized that driver fatigue can be detected early enough to avoid a car accident by looking at it. The detection of eye developments and flicker designs in an arrangement of pictures of a face is part of the exploration of fatigue. Initially, we decided to use Matlab to identify eye flicker designs. Mathematical control of power levels was used as the tool. The following was the calculation that was used.We started by using a webcam to capture a facial image. Binarizing the image was the first step in the preprocessing process. The top and sides of the face are differentiated to determine the zone where the eyes are located. Facial focus is found using the side of the face. It is used as a source of perspective when assessing the left and right eyes. The flat center point on the facial area is determined by descending from the highest point on the face.To characterise the eye region, major changes in the midpoints were used. When the eyes were closed, there was no difference in the flat natural, which was used to detect a squint.

Matlab, on the other hand, did have some real limitations. Matlab had extremely strict preparing requirements. Similarly, there were a few speed handling problems. Matlab was only capable of processing 4-5 edges per second. This was even lower on a platform with little RAM. Face flickering takes milliseconds, as we all know. A driver's head may also grow at a rapid rate. Even though our Matlab software observed an eye squint, the exhibition was found to be badly lacking.OpenCV was brought in to help with this. OpenCV is a PC vision library that is free and open source. It is optimized for computational efficiency by emphasizing ongoing activities. It allows fast and trouble-free production of refined vision applications. OpenCV met our application's criteria for low handling force and high velocity.

To differentiate between the face and the eyes, we used the Haartraining applications in OpenCV. Provided a set of positive and negative instances, this results in a classifier. The following methods were used:

a. Compile a database of face and eye information. These should be contained in at least one catalogue that is sorted by a book record. For the classifier to work well, it requires a large amount of high-quality data.

b.A vector yield file is generated using the utility programme createsamples().

We can rehash the preparation procedure using this record.

It quotes positive examples from pre-normalized images and changes their size to the specified width and height.

c.The Viola Jones course determines if an object in a photograph is similar to the preparation sequence. A negative illustration can be created from any image that does not include the object of interest. To gain proficiency with any article, consider the example of a negative foundation image. All these negative foundation images are put in a record, which is then listed.

d.Boosting is used to complete the image training. We familiarise ourselves with the gathering of classifiers one by one as we prepare. Every racer in the crowd is a powerless racer. These weak aristocracies are generally formed from a single variable choice tree known as the stump. To set up a favorite stall, use its information to determine team selection and also learn the burden for its election based on the accuracy of the information. Information is focused between the preparation of each individual rating, so more attention is paid to information focused on where the error is. This measure continues until a complete error on the data set has emerged.

e. This estimation is feasible when a large amount of preparing data is available.

f. Face and eye classifiers are needed for our project. As a result, we used object learning techniques to create our own .xml record for hair cutting tools.

g. About 2,000 positive examples and 3,000 were taken. Organizing them is a periodic collection cycle. Face.xml and haarcascade-eye.xml records have finally been developed.

h. These xml files are simply used for object recognition. It can recognize a series of objects (for our situation and our eyes). Haarcascade-eye.xml is clearly designed for open eyes. As a consequence, when the eyes are closed, the framework can not distinguish between objects. This is an example of a flicker. When a squint lasts longer than 5 casings, the driver is considered slow and an alert is released

2. Literature review

[1]Many methodologies have been established to improve the accuracy with which driver fatigue frameworks are identified.Mardi et al. proposed a sleepiness detection model based on electroencephalography (EEG) signals.To differentiate between laziness and planning, segregated turbulent highlights and the sign's logarithm of energy are omitted.A fake neural organisation was used for order, with an accuracy of 83.3 percent.To detect laziness, Noori et al. proposed a model based on a combination of Driving Quality Signals, EEG, and Electrooculography.

[2] Danisman et al. devised a technique for identifying laziness based on variations in the rate at which the eyes squint.Viola Jones recognition calculation was used to classify the face region from the images.A neural organization-based eye finder was then used to find the understudies' location.It was determined that the driver was sleepy if the number of flickers increased with each squint.Abtahi et al. devised a system for detecting sluggishness based on yawning.This technique begins by identifying the face, then distinguishing the eye and mouth districts.They discovered that the mouth opens as a result of a large mouth opening.The face with the largest opening shows a yawning mouth.

[3]CNNs is used by Dwivedi et al. to construct a model for evaluating the language.Convolutional neural organisation was used to capture inert highlights, followed by a SoftMax layer for characterization, resulting in a precision of 78 percent.To eliminate street mishaps caused by slow drivers, Alshaqaqi et al. proposed a Progressed Driver Assistance System.A calculation is proposed here to locate, follow, and examine the face and eyes in determining PERCLOS for det. The eye region was determined after identifying the face, and the Hough adjustment for circles (HTC) technique was used to assess the eye condition.If the eye is closed for more than 5 seconds, it is said to be tired.Park et al. suggested a deep learning-based organisation for detecting driver indifference from recorded data.Three prominent organisations, such as Alex Net, VGG-FaceNet, and FlowImageNet, are used to provide learning.The NTHU driver fatigue video dataset was used in this analysis, and the accuracy was about 73 percent.Using a Hidden Markov Model, Tadesse et al.developed a method for detecting driver sleepiness.In this work, the face districts were extracted using Viola Jones' estimate.The highlights in the face districts were extracted using Gabor wavelet decay.Highlights were chosen using the Adaboost learning equation.This is a valuable method for drained or non-languid articulation.An eye-following-based driver language system was implemented by Said et al. In this situation, the machine detects the driver's language and sounds a warning to alert him.In this project, the Viola Jones model was used to define the face region and eye district.In indoor tests, it had an accuracy of 82 percent, and in outdoor temperature tests, it had a 72.8 percent accuracy.For understanding driver exhaustion, Picot et al. used both visual movement and mind actions.The operation of the cerebrum was tracked using a single channel EEG.Squinting and representation are used to track the visual action.Squinting highlights are reduced using EOG.An EOG-based finder was created by combining these two highlights with fluffy logic.The accuracy of this study was 80.6 percent when it was checked on a dataset of twenty individual drivers.Mandal et al. developed a method for detecting fatigue in transport drivers based on dreams.In this analysis, a HOG and SVM are used separately for head-shoulder location and driver identification.When a driver was identified, they used OpenCV face identifier for face recognition and OpenCV eye finder for eye location.The ghost is fading away.Embedding was used to familiarise ourselves with the eye shape, and another technique was used to assess eye transparency.The highlights from two eye locators, I2R-ED and CV-ED, were combined to form a single image.PERCLOS was chosen for languor

recognition.Jabbara et al. suggested a model focused on profound learning for detecting driver sleepiness in Android apps.They came up with a model focused on the discovery of facial landmarks.The photos were extracted from the video outlines first, and then the milestone organise focuses were removed using the Dlib library.These dates serve as a pledge to a multi-year strategy.These benchmarks will aid in the creation of a multi-facet perceptron classifier.The classifier divides the population into two classes based on these focuses: sluggish and non-sluggish.This technique had an accuracy of more than 80% on the NTHU Drowsy Driver Detection Dataset.

3.Methodology Suggested

3.1 Algorithm

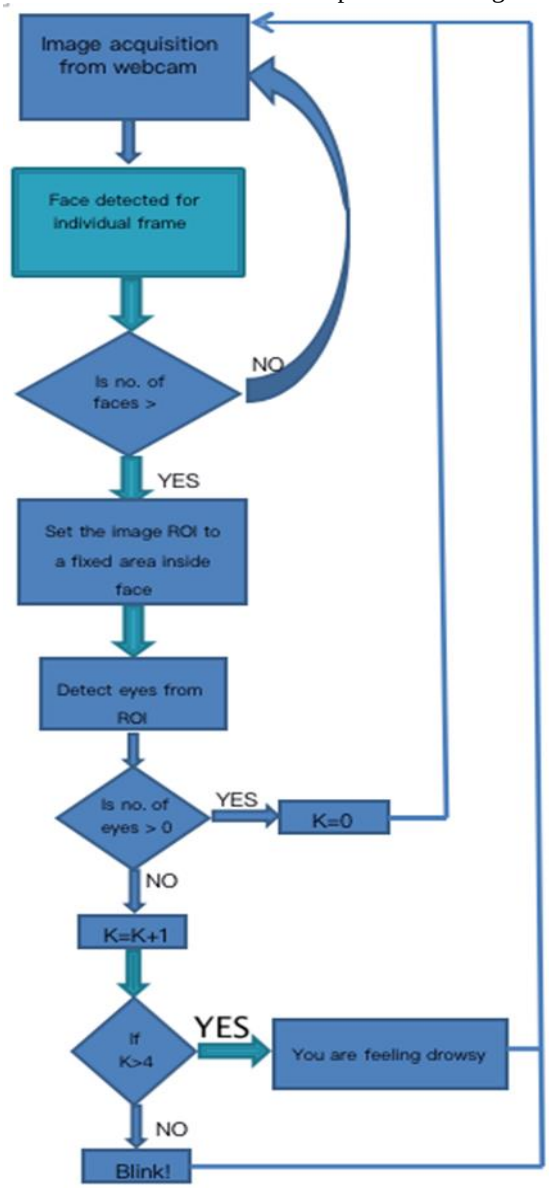
The length of time an individual's eyes are closed may be used to determine their languor.In our framework, the quicker identification and training of information is a crucial factor.

It is counted how many edges the eyes are closed on.A alert message appears on the dashboard if the number of edges exceeds a certain level, signalling that the driver is sleepy.

It is counted how many edges the eyes are closed on.The face.xml file from the Haar Cascade record is then used to analyse and distinguish the appearances in each individual edge.A new casing is obtained in the event that no face can be detected.If a face can be identified, an area of interest within the face is defined.This intriguing location is where the eyes can be located.When a place of interest is stated, the system's computational requirements are greatly reduced.Haarcascade eye.xml is then used to identify the eyes from the region of interest.

When an eye is detected, there is no flicker, and the squint counter K is set to '0.'When the eyes are closed in a specific casing, the flicker counter is increased, and a squint is detected.It's fair to say that the driver is exhausted if his or her eyes are closed on some edges.As a result, laziness is identified and an alert is given.

The entire interaction is then replicated as long as the driver maintains control of the vehicle.



3.2. Image acquisition:

For analysing a video transfer from the camera, cvCaptureFromCAM assigns a capacity and applies the CvCapture framework.It will be necessary to register the camera that will be used.If only one camera is available, or if the camera is used anyway, - one can be moved.Sets camera properties, for example, with cvSetCaptureProperty.We can set distance, stature properties, and recover outline using the CV CAP PROP FORMAT work, which returns -1.cvQueryFrame() takes a frame from a camera or video recording, decompresses it, and returns it.Grab Frame and Retrieve Frame are combined into a single call with this capacity.It is not appropriate to deliver the picture that has been returned.

3.3. Face detection:

The course is stacked by: We must use ->instead of since we have a pointer to the CvHaarClassifierCascade.Furthermore, the pointer is never instanciated, which is the opposite of what should be the case.

The grayscale image in CvArr is a grayscale image.The capability will be dependent on the position of a given region of interest (ROI).In this vein, using ROI to reduce image size constraints is one way to speed up face recognition.To create the classifier course, the Haar highlight course was jam-packed with cvLoad() in the face identify code.The capacity contention is an OpenCV calculation work buffer that is generated with cvCreateMemStorage(0) and then cleared for reuse with cvClearMemStorage(0) in the face discovery code (storage).The cvHaarDetectObjects() function searches the information picture for faces at all scales.Setting the scale factor boundary determines the size of the bounce between each scale; a higher value means quicker measuring time at the expense of possible missed recognitions if the scaling misses countenances of specific sizes.The minimum neighbours boundary protects against false detection.Since the surrounding pixels and scales often show a face, real face areas in a photograph will typically receive several "hits" in the same place.When this is set to the default (3) in the face identification code, we can choose a face if there are at least three covering discoveries in a given area.The banners boundary has four valid settings, which can be combined using the Boolean OR administrator (obviously).The first is CV HAAR DO CANNY PRUNING.When banners are set to this value, the classifier skips fl at locales (no lines).CV HAAR SCALE IMAGE is the next potential banner, which tells the calculation to scale the image rather than the finder (this can yield some presentation benefits as far as how memory and store are utilized).OpenCV is told to only return the largest item found by the CV HAAR FIND BIGGEST OBJECT banner option (thus the quantity of articles returned will be it is possible that one or zero).The last banner is CV HAAR DO ROUGH SEARCH, which is used in conjunction with CV HAAR FIND BIGGEST Item.In whatever scale the key competitor is found, this banner is used to put the chase to a close (with enough neighbours to be considered a hit).The smallest district where a face can be found is the final boundary, min scale.Setting this to a higher value will reduce calculation time but result in the loss of small faces.Gets three square components of the known face and returns them:

The rectangular area is now used to attract attention.

- Face in top left position (x, y)
- Length
- Height of the individual

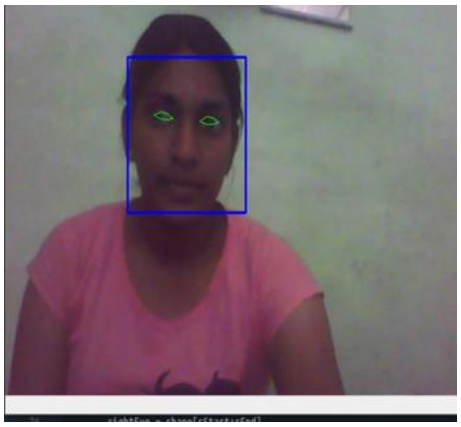
The upper left corner of the square shape is determined by the picture from which it should be drawn, which is cvPoint ((rx, ry)) (x+ width), (y+ stature): characterises the square shape's right base corner.

The objects in the image have been grouped together and saved in a rectangular area of the image. The square's form is determined by the areas returned by face location.In the above work, the picture for the relevant corner focuses in a square shape.The other boundary in the square form is for drawing line tone, thickness, and type.

3.4. Create a region of interest in the image:

It sets the ROI for the image accordingly. Within the picture, the square shape zone must be placed. The objective image must then be developed.cvGetSize will return the ROI's width and height.The picture is copied with cvCopy.With cvReset, the Area Of Interest will be reset.We took a few pixels from the face's leftmost point to a few pixels from the middle to half of its stature.The district and the face are the same distance.The rectangular region is now used to draw eyes.

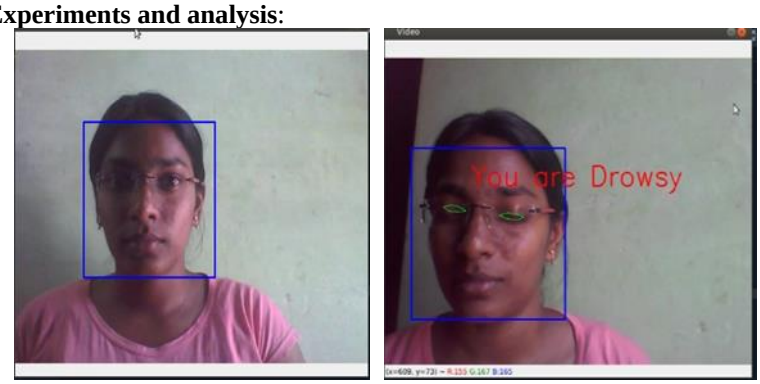
Eye detection:



lt has the same capabilities as face recognition. To distinguish eyes of different sizes in the photo, the eye course is used, and the smallest size of the article is shrunk and advanced. cvRectangle () is used to create a square shape for groupings of e yes located in a given edge, as previously explained. With cvResetlmageROl(img), the ROI is reset, allowing the entire image to be viewed in a window with the help of cvshowlrnage ("viden ", img). Since the course is only built for open eyes, when eyes are clv in any case, they are not automatically distinguished. Squint is the end state of the eyes, and the l anguor condition is shown using the c utText () function if the end state lasts for more than 7 consecutive edges. You may also use the community record order system(abc.mp3) to play any sound document when sluggishness is detected.

Before pressing a key, the location loop is repeated. When the key is pressed, the software uses the capacities specified below to stop the recognition capability, stop the display, release memory, and close the video window.Create memory for counts,Allocate memorystockpiling and Clear memory stockpiling when utilized previously

4.Experiments and analysis:



Each volunteer was asked to blink 20 times and become drowsy 6 times during the testing process. The accuracy for eye blink was calculated by the formula,

Accuracy = 1 - |total no. of blinks -no. of blinks detected| / total no. of blinks.

The same formula was used for calculating accuracy of drowsiness detection.

It can be seen from the result into consideration then the system has an accuracy of nearly 100%. In sample with out the backlight of the webcam, the resulting poor lighting conditions gave a highly erroneous output.

5.Future works:

When a driver's fatigue level reaches a certain breaking pnint, the ongoing driver weariness location system derilands that the vehicle be stopped. lt is recommended to schedule a clear scale driver weariJss recognition system rather than relying on edge sleepiness levels. It continuously tracks the amount of langu or and sends a signal to the vehicle's water-driven stopping mechanism when it reaches a certain threshold.

Hardwar elements are needed to assist with the hydraulic braking system, which includes a relay, timer, stepped motor, and linear actuator, as well as dedicated hardware for image acquisition and display. When the level rises above a certain threshold, a signal is produced and sent to the handoff through the equal port (parallel informa n move needed for quicker results). The on postpone clrck is activated by the handoff, which then triggers the stepped engine for a particular time period1. The stepper motor is connected to a straight actuator.

The direct actuator nverts the stepper engine's rotational production to straight movement. This straight movement is used to drive a shaft that is directly linked to the vehicle's water-powered braking system. The brake is applied as the shaft passes, and the vehicle's speed decreases. . Since it reduces the vehicle's speed to a safe

level, the chances of a crash are greatly decreased, which is extremely useful for avoiding drowsy-driving accidents.

6.Conclusion:

Therefore, with OpenCV, we have successfully designed a prototype for drowsiness detection. The developed system is successfully tested, its limitations identified and future action plan developed.

References:

1. Amodio, A., Ermidoro, M., Maggi, D., Formentin, S., Savaresi, S.M. (2018). Automatic detection of driver impairment based on pupillary light reflex. IEEE Transactions on Intelligent Transportation Systems, pp. 1-11. https://doi.org/10.1109/tits.2018.2871262
2. Yang, J.H., Mao, Z.H., Tijerina, L., Pilutti, T., Coughlin, J.F., Feron, E. (2009). Detection of driver fatigue caused by sleep deprivation. IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans, 39(4): 694-705. https://doi.org/10.1109/tsmca.2009.2018634
3. Hu, S., Zheng, G. (2009). Driver drowsiness detection with eyelid related parameters by support vector machine. Expert Systems with Applications, 36(4): 7651-7658. https://doi.org/10.1016/j.eswa.2008.09.030
4. Mardi, Z., Ashtiani, S.N., Mikaili, M. (2011). EEG-based drowsiness detection for safe driving using chaotic features and statistical tests. Journal of Medical Signals And Sensors, 1(2): 130-137. https://doi.org/10.4103/2228-7477.95297
5. Noori, S.M., Mikaeili, M. (2016). Driving drowsiness detection using fusion of electroencephalography, electrooculography, and driving quality signals. J Med Signals Sens, 6: 39-46. https://doi.org/10.4103/2228- 7477.175868
6. Dang, K., Sharma, S. (2017). Review and comparison of face detection algorithms. 7th International Conference on Cloud Computing, Data Science & Engineering, pp. 629-633. https://doi.org/10.1109/confluence.2017.7943228
7. VenkataRamiReddy, C., Kishore, K.K., Bhattacharyya, D., Kim, T.H. (2014). Multi-feature fusion based facial expression classification using DLBP and DCT. Int J Softw Eng Appl, 8(9): 55-68. https://doi.org/10.14257/ijseia.2014.8.9.05
8. Ramireddy, C.V., Kishore, K.V.K. (2013). Facial expression classification using Kernel based PCA with fused DCT and GWT features. ICCIC, Enathi, 1-6. https://doi.org/10.1109/iccic.2013.6724211
9. Krajewski, J., Sommer, D., Trutschel, U., Edwards, D., Golz, M. (2009). Steering wheel behavior based estimation of fatigue. The Fifth International Driving Symposium on Human Factors in Driver Assessment, Training and Vehicle Design, pp. 118-124. https://doi.org/10.17077/drivingassessment.1311
10. Danisman, T., Bilasco, I.M., Djeraba, C., Ihaddadene, N. (2014). Drowsy driver detection system using eye blink patterns. 2010 International Conference on Machine and Web Intelligence, Algiers, Algeria, pp. 230-233. https://doi.org/10.1109/icmwi.2010.5648121
11. Abtahi, S., Hariri, B., Shirmohammadi, S. (2011). Driver drowsiness monitoring based on yawning detection. 2011 IEEE International Instrumentation and Measurement Technology Conference, Binjiang, pp. 1-4. https://doi.org/10.1109/imtc.2011.5944101
12. Dwivedi, K., Biswaranjan, K., Sethi, A. (2014). Drowsy driver detection using representation learning. Advance Computing Conference (IACC), IEEE, pp. 995-999. https://doi.org/10.1109/iadcc.2014.6779459 [13] Alshaquaqi, B., Baquhaizel, A.S., Amine Ouis, M.E., Boumehed, M., Ouamri, A., Keche, M. (2013). Driver drowsiness detection system. 8 th International Workshop on Systems, Signal Processing and Their Applications (WoSSPA). https://doi.org/10.1109/wosspa.2013.6602353 [14] Park, S., Pan, F., Kang, S., Yoo, C.D. (2017). Driver drowsiness detection system based on feature representation

learning using various deep networks. In: Chen CS., Lu J., Ma KK. (eds) Computer Vision – ACCV 2016 Workshops, Lecture Notes in Computer Science, pp. 154-164. https://doi.org/10.1007/978-3- 319-54526-4_12

13. Tadesse, E., Sheng, W., Liu, M. (2014). Driver drowsiness detection through HMM based dynamic modeling. 2014 IEEE International Conference on Robotics and Automation (ICRA), Hong Kong, pp. 4003-4008. https://doi.org/10.1109/ICRA.2014.6907440
14. Said, S., AlKork, S., Beyrouthy, T., Hassan, M., Abdellatif, O., Abdraboo, M.F. (2018). Real time eye tracking and detection- a driving assistance system. Advances in Science, Technology and Engineering Systems Journal, 3(6): 446-454. https://doi.org/10.25046/aj030653
15. Picot, A., Charbonnier, S., Caplier, A. (2012). On-Line Detection of drowsiness using brain and visual information. IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans, 42(3): 764- 775. https://doi.org/10.1109/tsmca.2011.2164242
16. Mandal, B., Li, L., Wang, G.S., Lin, J. (2017). Towards detection of bus driver fatigue based on robust visual analysis of eye state. IEEE Transactions on Intelligent Transportation Systems, 18(3): 545-557. https://doi.org/10.1109/tits.2016.2582900
17. Jabbar, R., Al-Khalifa, K., Kharbeche, M., Alhajyaseen, W., Jafari, M., Jiang, S. (2018). Real-time driver drowsiness detection for android application using deep neural networks techniques. Procedia Computer Science, 130: 400-407. https://doi.org/10.1016/j.procs.2018.04.060
18. Viola, P., Jones, M.J. (2004). Robust real-time face detection. International Journal of Computer Vision, 57(2): 137-154. https://doi.org/10.1023/b:visi.0000013087.49260.fb
19. Jensen, O.H. (2008). Implementing the Viola-Jones face detection algorithm (Master's thesis, Technical University of Denmark, DTU, DK-2800 Kgs. Lyngby, Denmark).
20. O'Shea, K., Nash, R. (2015). An introduction to convolutional neural networks. arXiv preprint arXiv:1511.08458. https://arxiv.org/abs/1511.08458v2 [23] Kim, K., Hong, H., Nam, G., Park, K. (2017). A study of deep CNN-based classification of open and closed eyes using a visible light camera sensor. Sensors, 17(7): 1534. https://doi.org/10.3390/s17071534 [24] Lee, K., Yoon, H., Song, J., Park, K. (2018). Convolutional neural network-based classification of driver's emotion during aggressive and smooth driving using multi-modal camera sensors. Sensors, 18(4): 957. https://doi.org/10.3390/s18040957