Prototype of Driver Fatigue Detection System Using Convolutional Neural Network

Kseniia Nikolskaia[1], Vladislav Bessonov,
Artem Starkov
South Ural State University
Chelyabinsk, Russia
[1]nikolskaya174@gmail.com

Aleksey Minbaleev
Department of Information Law and Digital Technologies
Kutafin Moscow State Law University
Moscow, Russia
alexmin@bk.ru

***Abstract*—Driver fatigue is one of the major causes of accidents in the world. Detecting the drowsiness of the driver is one of the surest ways of measuring driver fatigue. In this paper we aim to develop a prototype drowsiness detection system. Presents a method for detecting the early signs of fatigue/drowsiness during driving. Analysing some environmental variables, it is possible to detect the loss of alertness prior to the driver falling asleep. This system works by monitoring the eyes of the driver and sounding an alarm when he/she is drowsy. As a result of this analysis, the system will determine if the subject is able to drive.**

Keywords—component; formatting; style; styling; insert (key words), Deep learning; CNN; (HOG); (Inattention;) Fatigue; Drowsiness;

I. Introduction

Modern cars take not only an integral part of people's lives, but are also one of the most dangerous types of vehicles. According to official sources, around the world every year, as a result of an accident, about 1.2 million people die, or 3,287 people per day [1]. One of the most common causes of death from road traffic accidents is driver drowsiness while driving and this is the cause of every fifth traffic accident. Therefore, the most dangerous condition of a person is overwork, which leads to loss of concentration and inability to drive a vehicle. Mistakes related to fatigue can cost human lives. We believe that ensuring the safety of people's livelihoods is one of the highest priorities in the world today. Leading experts in the automotive industry equip their cars with advanced technologies that reduce the influence of the human factor while driving. All sorts of sensors replace the role of human senses and uniquely interpret various situations on the road, choosing (in most cases) a rational strategy of behavior in an emergency. However, the work of the above-mentioned technologies is not yet perfect and requires further development. The relevance of the stated theme is indicated by the fact that every year car manufacturers present more and more advanced systems.

II. Review Analogues

One of the best examples of such systems can rightly be considered the Mercedes-Benz Attention Assist system. This system monitors the behavior of the driver, analyzing the following data: the driver's action while driving, steering, the way to drive the car. Schematically, the system consists of a steering wheel sensor, a warning light and an audible warning to the driver. The rudder sensor determines the force exerted on the steering wheel during its rotation and its change. In addition, the system takes into account the readings of other control sensors of the vehicle: the braking system, driving stability, engine parameters and visibility limitations. The result is determined by the method of installing violations in the actions of the driver and changing the direction of movement of the car. A signal with a soundtrack is sent to the dashboard screen informing the driver to stop for a rest. In case of ignoring the warning, when the driver in a sleepy state does not stop driving, the system continues to signal every 15 minutes [2].

Similarly to the Mercedes-Benz company, work on monitoring driving care with the help of computerized systems is carried out by the well-known company Volvo. Volvo's Driver Alert Control system differs from Attention Assist control in that it only determines the path of the car on the road, and the video control (which is set in the direction of the vehicle) determines its location on the roadway. If there is a departure from the established boundaries, the system responds to this as signs of driver fatigue. Two types of warning signals are issued: “hard” and “soft”, which depend on the driver's overall well-being. Signals differ in tone and volume [3].

The driver’s fatigue rating is assessed by the controlling scheme established by General Motors. In this scheme, the base is the tried and tested technique of Seeing Machines, used both on cargo transport and railway, as well as in quarrying. Specially built-in unit produces control of the driver’s eyes and concentration. The system gives a command to stop the movement of the vehicle. Detecting signs of fatigue or a state close to sleep and loss of driver care [4].

III. Face Detection in Images

One of the main tasks of identifying signs of loss of human concentration in the image is to determine the location of the face. Several public libraries are suitable for this purpose. In this paper, we decided to use the public dlib library. This library provides the ability to search for faces in an image using the HOG (Histogram of Oriented Gradients) algorithm [5]. The basic idea of the HOG algorithm is to assume that the appearance and shape of a face in an image section can be described by the distribution of intensity gradients. This algorithm works as follows. The original image is converted to

978-1-7281-2595-4/19/$31.00 ©2019 IEEE

grayscale. Further, for each pixel of the image, a gradient is determined in the context of neighboring pixels. In the next step, the image is divided into small squares of 16x16 pixels called cells. For each cell, a histogram of gradient directions is calculated, using the participation of each pixel of the cell in weighted voting for nine channels of the direction histogram. The cells are then grouped into larger square blocks, called descriptors, where the cell histograms normalize. Each cell can appear in more than one descriptor. The final step is to feed the descriptor to the pre-trained SVM classifier [6]. Based on this algorithm, a previously trained neural network called shape_predictor_68_face_landmarks.dat works. The work of this network is to search for the image of a human face and 68 key points on it (Fig. 1).

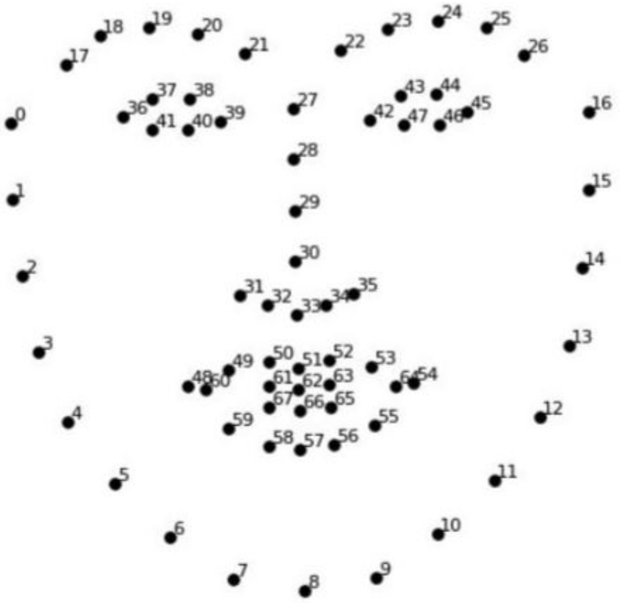


Fig. 1. Key points of HOG

IV. Algorithm to Determine the Signs of Losses of Concentration by Key Points

In order to increase the accuracy of the application, in addition to the CNN, we decided to develop a set of algorithms for recognizing signs of loss of concentration in humans on the image using as input data the result of the operation of the neural network shape_predictor_68_face_landmarks.dat, which represent 68 key points on the human face. Each of the algorithms is responsible for a separate indication of loss of concentration. Of all the signs of loss of concentration, the main four were identified: the eyes closed, the driver's gaze distracted from the road, the head position not in the direction of driving, and the yawning as a warning device for fatigue and drowsiness in the driver.

Consider an algorithm for recognizing closed eyes. The main criterion that the eyes of a person in the image are closed is the closure of the eyelids. Of the 68 key points beyond the borders of the eyes (eyelids), the following points are responsible: 36, 37, 38, 39, 40, 41 for the left eye and 42, 43, 44, 45, 46, 47 for the right eye. Figure X shows key points for analyzing the closed eyes of a person in the image

We calculate the ratio of the length of the obtained segments using their lengths. This value is the most accurate criterion for determining closed eyes.

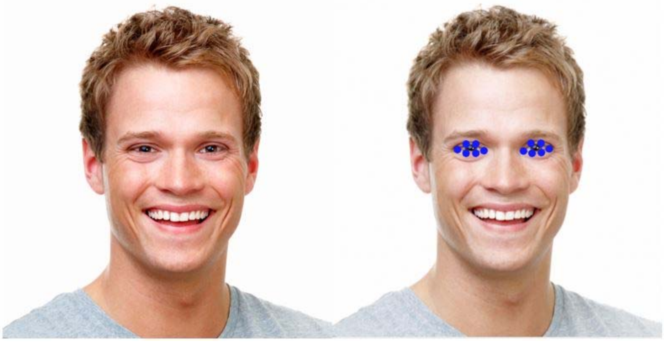


Fig. 2. Key points for analyzing closed eyes

To determine whether the eyes are closed, we need to build two intersecting segments: horizontal - with the ends on the inner and outer corners of the eye (axis of the palpebral fissure) corresponding to 36 (42) and 39 (45) key points, and vertical - with the ends in the middle of the upper and lower eyelid, corresponding to the middle between 37 and 38 (43 and 44) and 40 and 41 (46 and 47) (Fig. 2).

V. Implementation of Neural Network and Input Data

The most acceptable solution for the problem of effective image recognition is convolutional neural networks - a special architecture of artificial neural networks that helps reduce the feature vector and contains 3 main paradigms: local perception, shared weights and sub-sampling. The issues of memory, performance and training time are always relevant when using neural networks. Fig. 3 shows the architecture of the driver fatigue CNN topology [7].

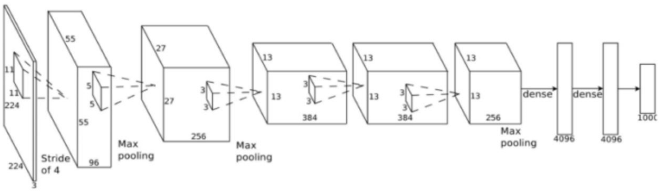


Fig. 3. Architecture of the Driver Fatigue CNN detector

Local perception implies that not the whole image is fed to the input of one neuron, but only some of its area. This approach allowed us to preserve the topology of the image from layer to layer. Shared weights suggest that for a large number of links a very small set of weights should be used. For example, if there is a 28x28 pixel image at the input, each of the neurons of the next layer will receive only a small section of this image of 3x3 in size, and each of the fragments will be processed with the same set of weights. The introduced weight limit improves the generalizing properties of the network, which ultimately has a positive effect on the network performance. The purpose of subsampling is to reduce the spatial dimension of the image. Initial image is reduced by a specified number of times. Subsampling, in turn, is necessary to ensure scale invariance. Preliminary image processing and the correct choice of neural network architecture are necessary to achieve high accuracy of recognition of the system and improve its performance. The input data for the neural network are images of any size in the format .jpg or .png. We decided not to use the noise reduction procedure in the image preprocessing process, since we use a CNN, which, due to the use of the concept of shared weights, has the ability to respond mainly to images.

The selected topology of the neural network was chosen empirically. It contains 3 convolutional layers, 3 subsampling layers, 1 regularization layer, and 2 fully connected layers. Several topologies were tested with a different number of hidden layers, however the selected topology showed the best result among all tested. The purpose of the convolutional layers is to obtain a recognized image description in the form of a set of feature maps. The purpose of downsampling layers is to reduce the dimension using the max-pooling method — the entire attribute map is divided into cells, from which the maximum values are selected. The regularization layer is a way to cope with the overfitting of a neural network. Each neuron is connected to all neurons in fully connected layers at the previous level, with each connection having its own weighting factor. The input data for the neural network is an image of 100x40 pixels in digital form (two-dimensional array of pixels). The output of the neural network is a binary number, where 0 - the driver shows no signs of fatigue, 1 - the driver gives signs of fatigue.

In order to improve the accuracy of the neural network used in the system for determining the concentration loss of the driver, it is necessary to carry out the initial data processing, transferring them to the format required for the neural network and eliminating unnecessary information from them. The input data is a video stream, received either from a built-in video recorder (webcam) in real time, or a pre-recorded video file. The first preprocessing stage is to translate the input video stream into an image sequence. The OpenCV library is used to get a video file and its subsequent splitting into frames. The second stage of preprocessing consists in determining the face of a person on the image [8]. The third stage is to select the eye area in the previously selected area of the image with a human face and resize this area in accordance with the requirements for the input data of the neural network. The fourth stage in preprocessing is resizing a previously selected image of the eye area. The result of the preprocessing is the image of the eye area of 100x40 pixels.

The Keras library is selected to create and train a CNN model. This library has convenient tools for creating a neural network model according to the previously selected topology, and its further training. he network was trained in 147 epochs on the NVIDIA GeForce GTX 1050 video card. Training time was 1 hour 45 minutes. The accuracy of the classification, calculated as the ratio of the number of images for which the classifier made the right decision and the total number of images in the sample, turned out to be 89% on the test sample.

The preparation of a multitude of training data is crucial for successfully solving machine learning problems. The tasks of machine learning often consist in the correct formation of a training set. Due to the active development of neural networks, the problem of forming a training sample is very relevant, since in many tasks deep neural networks demonstrate results that are significantly superior to other machine learning algorithms. At the same time, in the modern machine learning literature, the issues of formation of the training set are completely ignored or insufficient attention is paid to them. We decided to use the "Closed Eyes In The Wild (CEW)" [9] dataset for training. This data set consists of 2,500 photographs with the faces of people depicted on them with eyes closed and open. The size of the training and control samples amounted to 2,500 images, divided into 2 different classes with open and closed eyes.

VI. Implementation of Loss Recognition System Concentration

In order to identify signs of fatigue and loss of concentration in the driver, in addition to the self-trained and tested CNN, we decided to use the previously trained neural network model shape_predictor_68_face_landmarks.dat. With this model and the dlib library, we can mark the face of a person on the image with 68 key points. For convenience, the math library is used in computational work with the coordinates of key points. Each algorithm is software implemented as a separate function. Each function takes as input the coordinates of the key points needed for the calculations in this function and returns True as the result if the signs of loss of concentration were detected, and False otherwise. The main function responsible for receiving data, processing it and issuing the result uses all the implemented functions of recognizing signs of loss of concentration in humans and processes the results of these calculations.

The task of the concentration loss recognition system is to read the data for analysis, process the received data, as well as analyze them in order to detect signs of loss of concentration and display the processing result in case of a positive verdict. The software was implemented using OpenCV libraries for correct reading of the input video stream and its subsequent splitting into frames, preprocessing data for the neural network and displaying the results of the system; dlib for recognition on the image of a person's face with the allocation of key points for further processing by algorithms for recognition of signs of fatigue and loss of concentration; math for convenient work with calculations in algorithms. The main program thread, which is responsible for drawing the interface, reading the video file and its further processing and analysis with displaying the result to the user, contains two main functions, in addition to drawing the interface. These functions are responsible for the correct reading of the video stream and the analytical processing of the read files. User is provided with a choice of two ways to use the program. The first option is to select a pre-recorded video file for analysis. In this case, start the implementation of the video reading function, its subsequent division into frames and sequentially transmitting the frame unit analytical processing. The PyQt5 [10] and OpenCV libraries, respectively, are responsible for selecting the file and its subsequent reading. The second function performs the task of reading a video stream from a recording device in real time, its subsequent splitting into frames and the implementation of analytical processing of the read video file being divided into frames. The OpenCV library is responsible for reading the output stream. Both of the above functions perform the same task for different options for obtaining input data by the program. The last stage of both functions is the processing of frames received as a result of splitting the video file and displaying the results to the user. After splitting, each frame is analyzed by a face detector, which is a built-in function of the dlib library. After that, on the face found by the detector, 68 key points are determined using a pre-trained neural network

The closed eyes in the image is determined in two ways. The first method is to use a self-implemented and trained CNN. For this purpose there blinking_recognition () function. This function takes as arguments the image, the coordinates of the upper left vertex of the rectangle containing the face defined by the detector, and the length of the sides of the rectangle. First of all, the function pre-processes the resulting image, highlighting only the eye area for the neural network, which is 40% of the height of the rectangle with the top indent of 10% and the bottom in 50%. The resulting eye area is converted to a black and white image, and its size changes in accordance with the requirements for the correct operation of the neural network model, that is, 100x40 pixels. After that, the neural network model analyzes the image of the eye area obtained as a result of preprocessing and gives the result, the value of which is the class to which the neural network assigned the given image (0 - eyes closed, 1 - eyes open). In accordance with the result of the neural network operation, a variable will be created, which will be assigned the value True if the closed eyes were recognized and False otherwise. The second way to determine the closed eyes of a person in an image is to use key points defined by the neural network model shape_predictor_68_face_landmarks.dat. The function takes as input three arguments - two array indices of points required to determine the closed eyes and a full set of key points. The criterion for closing the eyes is the closeness of the eyelids, which, in turn, is characterized by a small distance between the middle points of the lower and upper eyelids of both eyes. However, it was experimentally established that determining the distance between the middle of the eyelids is not enough to correctly determine the closed eyes as a result of a wide variety of eye sizes and shapes in different people. For the most accurate determination of this, it is necessary to calculate the ratio of the height of the eye (the distance between the middle of the eyelids) to its width (the length of the palpebral axis). Determining the height and width of the eye is similar to determining the distance between the upper and lower lips of the algorithm discussed earlier. First of all, it is necessary to determine the ratio of the width and height for the left and right eyes, and then determine the arithmetic average of these relations. Then, the obtained result should be compared with the threshold value for closed eyes, calculated automatically when testing the work of determining the ratio of the height and width of the eye on the examples of images with people with different eye condition. The function returns the value True if the eyes are closed on the image and False otherwise.

VII. Application Implementation

An activity diagram was developed to implement the system (Fig. 4). Thad diagram allows us to describe the logic of procedures, business processes and workflows [15].

When the user uploads an image to the program, sequential preprocessing of the image begins, searching for the face and eye area. If the face is not found, an error message will be displayed in the output of the program. After preprocessing, the driver will automatically detect the loss of concentration in the driver in the image. After that, the neural network makes a decision, which will be displayed in the result field of the program.

We decided to use a CNN model that solves the problem of image classification with faces of people with closed and open eyes, and algorithms for determining loss of concentration in humans using the previously trained neural network model shape_predictor_68_face_landmarks.dat. to determine the signs of loss of concentration in the driver.

The main function of the system first draws the interface. After the user selects a data source, either the file is read from the specified directory, or the video stream is recorded from the webcam. Further there is the video splitting into frames, then the pre-processing function is used. As a result we get image of eye area of 100x40 pixels. The raw frame is analyzed by functions of concentration loss based on the model of a previously trained neural network model shape_predictor_68_face_landmarks.dat. After preprocessing, the frame is analyzed by a trained neural network. All video file frames are processed in this way. In case when at least one of the functions returns a True value (except for the yawning recognition function) for three seconds, the user is notified that there are signs of loss of concentration. If the yawning recognition function gives a positive value at least three times over a period of one minute, a message is displayed indicating that signs of fatigue and drowsiness are detected.

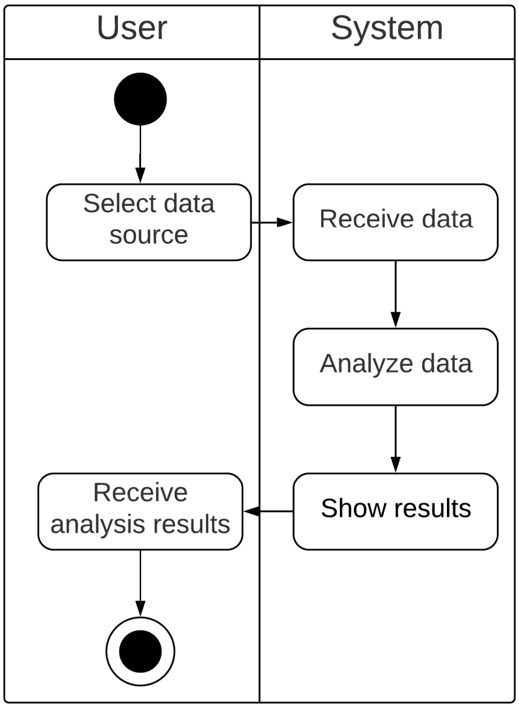
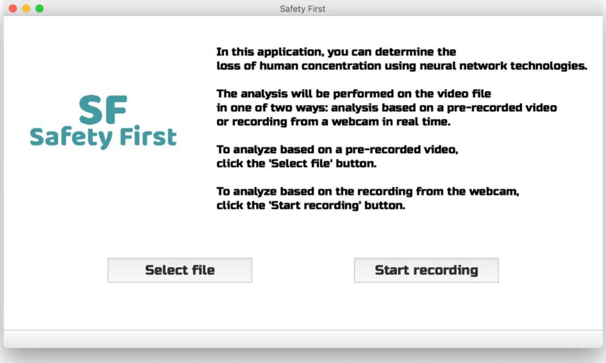


Fig. 4. System activity diagram

We developed a graphical interface for comfortable user interaction with the system. For the development of the interface, we used the PyQt5 graphic library and the QtDesigner application. Fig. 5 shows the final graphical user interface.



In this application, you can determine the loss of human concentration using neural network technologies.

The analysis will be performed on the video file in one of two ways: analysis based on a pre-recorded video or recording from a webcam in real time.

To analyze based on a pre-recorded video, click the 'Select file' button.

To analyze based on the recording from the webcam, click the 'Start recording' button.

Fig. 5. User Interface Appereance

ACKNOWLEDGMENT

The reported study was funded by RFBR according to the research project № 18-29-16015.

REFERENCES

[1] Analysis and forecast of the state and level of accidents on the roads of the Russian Federation and ways to reduce it. [Electronic resource] // URL:https://e-koncept.ru/2016/96251.htm (date of the application: 18.02.2019)

[2] Attention Assist [Electronic resource] // URL: https://www.mbusa.com/mercedes/benz/safety (date of the application: 29.04.18)

[3] Driver Alert Control [Electronic resource] // URL: https://www.media.volvocars.com/ru/ru-ru/media/videos/12261 (date of the application: 29.04.18)

[4] Driver Alert Control [Electronic resource] // URL: https://www.media.volvocars.com/ru/ru-ru/media/videos/12261 (date of the application: 29.04.18)

[5] Histogram of Oriented Gradients [Electronic resource] // URL: https://www.learnopencv.com/histogram-of-oriented-gradients/ (date of the application: 29.04.18)

[6] Support Vector Machines [Electronic resource] // URL: https://scikit-learn.org/stable/modules/svm.html (date of the application: 29.04.18)

[7] Finding Better Topologies for Deep Convolutional Neural Networks by Evolution Honglei Zhang, Serkan Kiranyaz, Moncef Gabbouj [Electronic resource] // URL: https://arxiv.org/pdf/1809.03242.pdf (date of the application: 29.04.18)

[8] OpenCV @ CVPR 2019 [Electronic resource] // URL: https://opencv.org/ (date of the application: 29.04.18)

[9] Closed Eyes In The Wild (CEW) [Electronic resource] // URL: http://parnec.nuaa.edu.cn/xtan/data/ClosedEyeDatabases.html (date of the application: 29.04.18)

[10] PyQt5 [Electronic resource] // URL: https://pypi.org/project/PyQt5/ (date of the application: 29.04.18)