

Sleepiness Detector Using a Webcam

Mirella López Picazo*, Lluís Cabet Carbó†

*Centre for Mathematical Sciences, Lund University, Sweden

Email: mirella.lopez.picazo@gmail.com

†Escuela Técnica Superior de Ingenieros de Telecomunicación, Universidad Politécnica de Valencia, Spain

Email: lluiscanet86@gmail.com

Abstract—The huge amount of traffic, domestic and industrial accidents due to low levels of alertness in people responsible of handling the systems that cause those accidents, has motivated a lot of researchers to find a technological solution to prevent them. The objective of this paper is to provide a software algorithm that will be able to monitor the level of alertness of drivers in order to aware them in case this levels are dangerously low. After analyzing the study of the art in this field, an approach based on the eye closing of the driver was chosen. Our algorithm is able to process low quality webcam images from the face of the driver being monitored to feasibly detect if the subject has had his or her eyes closed time enough to be dangerous for driving and provides a feedback to the driver in order to wake him or her up or let him or her know that he or she is in serious danger and should take measures to avoid the drowsiness. Although the algorithm works reasonably well it has some limitations as the use of sunglasses by the drivers or the use of the algorithm during nighttime because it is very sensitive to illumination levels.

I. INTRODUCTION

A low level of alertness due to sleepiness causes a lot of driving accidents. Drowsiness in drivers significantly deteriorates the capability to manage a car and it can be related to the lack of external stimulus, sleep, huge intake of food or drinks, specific illnesses or to the intake of some kind of medicines. Thus, the risk of an accident to themselves or other drivers is increases. The number of accidents has been incremented enough to become a serious problem for the society. In Great Britain, 20% of the accidents in highways were because of this reason [1]. A survey made in Buenos Aires to 770 drivers showed that 44% of them had frequently or very frequently felt drowsiness symptoms while driving and 45% of them had been very close to have an accident due to this reason. 66% of the drivers confessed that they stopped to sleep during 30 o 40 minutes and 73% of them did not reduce the speed [2]. A driver under the effects of fatigue decreases progressively driving attention and a loose of response capability against specific conditions that require immediate reactions while driving [3]. In the National Transportation Safety Borrad, fatigue was the cause of 30-40% of the accidents between 1991 and 1995. From them on fatigue has been recognized as one of the more relevant topics related to driving safety.

The objective of this paper is to provide a software algorithm that will be able to monitor the level of alertness of drivers in order to aware them in case this levels are dangerously low. But in order to have a good acceptance in the industry the system that uses our algorithm has

to meet the following requirements: easy to install, non invasive, easy to use during a long period of time, robust and real time algorithms.

In order to choose a feasible approach to monitor the sleepiness levels of a driver we have to analyze the state advantages and disadvantages of the ones that have been developed by other researchers so far. Three main approaches have been taken in this field through years. The first one is based in analyzing the mechanical parameters of the vehicle that would show a decrease in the level of alertness of the driver. Some studies showed that these parameters could be a decrease of the variations in the speed of the vehicle [4], [5], [6], increase of the amplitude of the steer wheel movements [7], [8] or a decrease of the shifting frequency in the gear shift [9],[10]. The second one is based in the human ocular behavior [11]. There are different variables that can be taken into consideration such as pupil size, blink frequency and duration or percentage of closed eyes in time (PERCLOSE). The third approach that can be taken is based on the analysis of the brain signals (EEG). [12] made a comparison of the previous approaches classifying the third one, as the most exact but most unpractical to develop real system and the third one is the less accurate but most easy to practically develop a system. Thus, we decided to take the second approach to develop a system with the specifications expressed previously, because of its trade off between accuracy and feasibility. The final prototype we got used a webcam for image acquisition because of its low cost and all the results shown below correspond to an implementation of the algorithm preformed in MATLAB.

II. ALGORITHM IMPLEMENTATION

Our algorithm is formed by two principal phases, initialization and execution, with several subphases. To make a general idea of the algorithm we can see the flow diagram at Fig. 1

A. Inicialization

The initialization is the first of two phases in which our algorithm can be divided, and we might consider it as the phase in which the system adapts to the new user. The tasks performed at this stage are the following:

1) *Video Device Inicialization*: We begin with the capture of images in real time using the webcam. We configure some properties of the incoming image, such

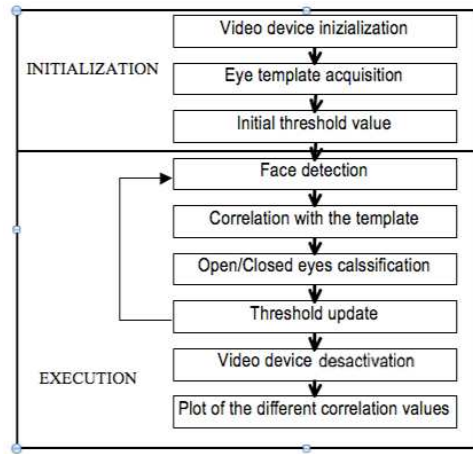


Fig. 1. Flow diagram of the algorithm

the parameter `TriggerRepeat` and the frequency with which the intervals of the frames that enter through the video will be recorded. Finally we extract the dates of the last image stored and process it with a face detection algorithm.

2) *Getting the Eye Template:* Our algorithm is the comparison of an open eye image with successive eye images that are going to be taken. At first we tried to use a standard eye template but did it not give us very good results when tested with different subjects as the correlation difference between open and closed eyes was not significant. So we chose to obtain a different template of each different user's eye, and thus obtain higher correlation values. For this, the first step is to detect the face and trim the portion of the image where the eye is.

For detecting the face we rely on an algorithm already developed: `fdlibmex`, created by Wolf Kienzle [13]. The algorithm is based on library calls that whose code we do not have access to, but after evaluating several versions of face detection algorithms, this was the one that performed the best. The output of this function is a matrix where each row corresponds to each of the detected faces and three columns. The first column gives the x coordinate of the center point of the square that surrounds the face, the second column y coordinate and the third the size of the side of the square. Once we have identified the faces we take the first detected face (first row) and cut out around the left eye (in the picture, the upper left quarter). To avoid errors in the detection of faces, we check whether the dimensions of the corresponding part are similar to the face dimensions that we have considered reasonable. Once this is checked we crop the part of the picture that only contains the eye. Following standard measures we are left with $2/3$ of the right side at x, and $1/3$ centered at y.

In the case of not detecting faces the output matrix will be empty and the detector has to warn the user to change his position.

It may be that due to user movement, to an error in

the detection of faces or some flicker that the image taken as a template is not very good and because of this the algorithm doesn't work accurately. Therefore, to avoid these problems we ask the user to hold a useful position to take another image of the eye template by showing a message on screen. After the user's response the algorithm will proceed to repeat the acquisition of the template or continue with the algorithm. Finally, we normalize the template by subtracting its average in order to calculate the cross-correlation with the following images obtained.

3) *Calculation of Initial Threshold:* Once the template is obtained, to discern whether the eye is closed or open in the next frames, we need to establish a correlation threshold with the template. One of the major problems that we had developing the algorithm was to choose which value would this threshold be because, depending on the individual and the light conditions, the values correlation could vary. Therefore, as a solution, we decided to get two new images with eyes open and calculate the correlation of the template with the average of both images.

B. Execution

Execution is the second phase of our algorithm, where we really detect the fatigue. We have created the algorithm to perform this detection for 30 frames and to save each one of these images in memory to compare them afterwards with the final plot that shows all the maximum values of the correlations for each of the images. The tasks performed in this part are following:

1) *Face Detection:* Just like when we found the template or initial threshold value.

2) *Calculation of Correlation with the Template:* When we have the new detected face we calculate cross-correlation between the template that we have obtained at the initialized phase the upper left quarter of the image of our face detected now. But this way we obtain a correlation value for each pixel of the image, thus we get too much information to decide whether the eye is open or closed in the new frame obtained. Therefore, we decided that among all these correlation values we pick the maximum that will correspond to a point belonging to the eye and we may indicate its resemblance to the template.

3) *Classification of Open/Close Eyes:* Afterwards, the algorithm makes a comparison of the new value of the maximum correlation with the threshold and if the correlation is higher it is decided that the eyes are open and if not, closed. If it detects that the eyes are closed, an alarm would sound in order to aware the driver and wake him up in case he was sleeping.

The value of the maximum correlations with eyes closed and open is changing in time due to changes in

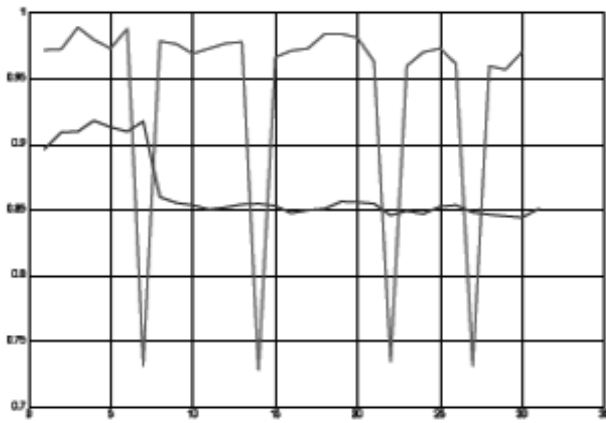


Fig. 2. Plot of the cross-correlation and adaptive threshold for each of the iterations

user position, changes in light, and so on. Therefore, it was not a good idea to have a fixed value of the threshold as this produces errors in long term, thus we decided to establish an adaptive threshold. To do this, when it detects a frame with opened eyes we update a variable where the maximum correlation value of the previous frame with opened eyes was stored. On the contrary, if eyes are closed we update a variable where the minimum correlation value of the previous frame with closed eyes was also stored. Then, after each iteration we update the threshold value by calculating the average between those two variables. However, sometimes it was found that some errors were still occurring in the first iterations and we decided not to set the alarm until we had obtained the first 4 frames and then used the 4 first values obtained to adjust the threshold to a more reasonable value.

4) Disconnecting Video Device and Correlations Plot:

The last step in our algorithm is to disable the video device and display a plot that shows the correlation value and the threshold for each iteration. In Fig. 2 we observe this plot, where in the X axis we have the number of the iteration and in the Y axis the value of the correlation. We realize that when having a frame with closed eyes there is a drop in the correlation. That happens in frames 7, 9, 22 and 27.

III. CONCLUSION

As we explained in the previous part, in we can observe a huge drop in the cross correlation between the template and the frames with closed eyes. However, this test have been performed with very good light conditions and with no light intensity changes, but due to the adapting threshold we dont expect to have many problems if the light changes occur within an reasonable interval.

The most important limitations of the algorithm are the following. Although it works properly when the subject is using glasses with transparent lenses, it is not useful in the case when the subject is wearing sunglasses, because no images from the eye can be obtained in this case. A second limitation is the fact that the face detection algorithm used is the best found but it is not working perfectly all the time, so a more robust algorithm should be find. Furthermore, our adaptive threshold is too dependent of the initialization, this means that if we dont have a good initialization value it may not converge to a useful value and the algorithm wont work properly. The algorithm initialization process could be uncomfortable for the user when applying it to a real system so alternative ways of making it more automatic should be considered.

Despite all these limitations exposed above, the authors consider that the results obtained with this new algorithm are satisfactory because we get a low rate of errors in reasonable conditions and this system works with a low cost webcam acquiring and it is designed to work with a low frame frequency acquisition, reducing the computational cost of the algorithm. Thus, this algorithm can be used to develop a real prototype at a cheapest cost than other that have been developed previously.

ACKNOWLEDGMENT

The authors would like to thank to Maria de la Salud Guillem Sánchez and Jose Millet Roig, professors of the course Biomedical Imaging and Signals in the ET-SIT, Universidad Politécnica de Valencia because they encouraged us to start this project, helping and guiding us during the whole development. The authors would also like to thank to the Centre for Mathematical Science of Lund University, specially to the professor Niels Chr. Overgaard, for their support to this project.

REFERENCES

- [1] H. J. et Al. Lancet, 1995.
- [2] Pérez-Chada, 2005.
- [3] B. ID., "Driver fatigue. human factors," MRC, Applied Psychology Unit, Cambridge, England, Tech. Rep. 36(2):298-314, 1994.
- [4] R. A. et Al., 1999.
- [5] C. F. et Al., 2000.
- [6] J. et Al., 2001.
- [7] C. B. et Al., 1993.
- [8] V. W.B. and Z. D.M., 2000.
- [9] W. et Al., 1992.
- [10] W. et Al., 2001.
- [11] Q. Ji and X. Yang, 2002.
- [12] S. L. C. S. . R. Alvarez, "bachelor thesis." Escuela Politcnica Nacional de Ecuador.
- [13] W. Kienzle, "fdlibmex — fast and simple face detection." <http://www.mathworks.com/matlabcentral/fileexchange/20976>, 05 Aug 2008 (Updated 02 Sep 2008).