Fuzzy logic approach for detecting drivers' drowsiness based on image processing and Video streaming

This paper was downloaded from TechRxiv (<https://www.techrxiv.org>).

LICENSE

CC BY 4.0

SUBMISSION DATE / POSTED DATE

08-10-2021 / 14-10-2021

CITATION

Yousif, Jabar (2021): Fuzzy logic approach for detecting drivers' drowsiness based on image processing and Video streaming. TechRxiv. Preprint. <https://doi.org/10.36227/techrxiv.16777927.v1>

DOI

[10.36227/techrxiv.16777927.v1](https://doi.org/10.36227/techrxiv.16777927.v1)

Fuzzy logic approach for detecting drivers' drowsiness based on image processing and Video streaming

Wasin AlKishrime^{1,*}, Abdallah Abualkishik², Mahmood Al-Bahriand³, Jabar H. Yousif⁴

^{1,*} Department of Computing and IT, Sohar University, Sohar 311, Oman; wkishri@su.edu.om

² Department of Computing and IT, Sohar University, Sohar 311, Oman; AAbualkishik@su.edu.om

³ Department of Computing and IT, Sohar University, Sohar 311, Oman; MBahri@su.edu.om

⁴ Department of Computing and IT, Sohar University, Sohar 311, Oman; jyousif@su.edu.om

* Correspondence: wkishri@su.edu.om;

Abstract: Driver drowsiness is a severe problem that usually leads to many traffic accidents and is classified as one of the most dangerous factors. Driver's response is sometimes delayed for various reasons and may not occur on time. Therefore, preventing and reducing driver drowsiness accidents is vital in a field of research that has received more attention due to the high rate of car accidents resulting from driver drowsiness. This paper aims to design a system that detects drowsiness in real-time based on image processing using fuzzy logic techniques. The approach focuses on various visual markers to detect the driver's level of drowsiness. Two main parameters are used to detect driver drowsiness: the duration of the eye blink and the state of the mouth. The Viola-Jones algorithm and image processing method are used to detect the image of the desired facial features in a specific driver video frame. The driver's face is detected and extracted extra features of the eyes and mouth areas. Start the analysis process to determine the status of these parameters. Second, the Kalman filter method is used to track and manipulate the difference in the size and orientation of the captured features. The technique checks all image states such as brightness, shadows, and clarity. Third, the blur control system provides different alert sounds based on the information tracked from the face, eyes, and mouth. The proposed method uses real-time data recorded by a webcam tool in the MatlabR2016a environment. The data sample contains videos of different users of different races, whether they wear glasses, gender, and various lighting backgrounds. The proposed system achieved an accuracy of up to 94.5% in the detection driver status.

Keywords: Driver drowsiness; face detection; eye-tracking; fuzzy logic; driver fatigue detection

1. Introduction

According to National Safety Council (NSC) [1], driver drowsiness is one of the significant factors that cause many traffic accidents in the world. Annually upwards of 100,000 vehicles are crashed, and around 1,500 people die. Annually, almost 12.5 billion dollars are the total loss cost for these types of accidents. Preventing such driver fatigue accidents is a high focus effort on many current types of research [2-4]. Driver fatigue increases the need to develop a monitoring system that analyses driver status and provides different alert sounds based on his facial features. Most researchers concentrated on detecting drowsy driving through analyzing the eyes' pupil's parameters [5-7]. Through these researches, driver drowsiness investigations are based on capturing the driver's video and detecting the driver's face using some technique. After that, they analyze the eye blinking frequency and decide the driver fatigue status [8]. Some other researcher [9-11], including the mouth, features too. For some reason, the system may work inefficiently to detect driver drowsiness. Varying light conditions and vibration of the driver is one of the main challenges to identifying the driver status in real-time [12]. The cloud computing techniques was deployed for car accident handling as in [13]. However, using multiple visual features and cues is more efficient for detecting driver drowsy. The proposed system detects driver fatigue in real-time through observing various facial features and select the correct driver state. It provides different alert sounds based on a different level of drowsing for the driver. This has been done by using a fuzzy logic technique in addition to the face features detection process [14]. This paper is organized as follows: Section 2 explores significant reviews of previous studies and related works in the same field of research systems. Section 3 describes the model of the system and how the experiments were conducted. Section 4 displayed the results of the conducted experiments. Finally, present the conclusion and future work.

2. Background and Related Work

2.1. Driver Drowsiness

Drowsiness causes significant social and economic losses to the country about road accidents that often occur on highways. Drowsiness accidents happen when the driver delays responding to a specific situation and lose control of the vehicle. Moreover, it is difficult to determine the level of the driver's drowsiness because it cannot be measured after the accident. Drowsiness usually affects various attitudes such as vigilance, decision-making, and concentration for drivers [15].

2.2. Driver Fatigue Monitoring Techniques

Various techniques of monitoring vigilance and fatigue are used to measure driver performance, including the following.

2.2.1 Physiological Behaviors

Researchers have categorized this technique as the most accurate method to detect fatigue because it is based on physiological measures such as heart rate, eye movements, respiration, and brain waves. This technique is effective when the electrical activity of the brain and body muscles is recorded. These parameters are collected from various sensors placed on the driver's body or embedded in the car. The driver usually wears a wristband to measure heart rate and a helmet or special contact lenses to monitor eye or eye movements. Despite its effectiveness, the main drawback of this technique is intrusive because it requires attaching electrodes to the driver's body, which causes the driver discomfort [14].

2.2.2 Indirect Vehicle Behaviours

This technique requires a significant amount of time to analyze user behavior. It includes lateral position, steering wheel movements, and time crossings, which indicate the driver's vigilance and fatigue level. It is categorized as a non-intrusive technique, but it has several limitations: vehicle type, driver experience, engineering characteristics, and road condition.

2.2.3 Directly Observable Visual Behaviours

People with fatigue show many observable behaviors usually observed in facial features, such as eye movement, head movement, and facial expression. The technique is based on different typical visual characteristics that detected from captured image for a drive. The captured image includes parameters, such as a lower level of deflection, longer luminescence, slow eyelid movement, a smaller degree of eye-opening, eye closure, repetitive gestures, yawning, and slow motion [14].

2.3. Related work

Various researchers use different methods and algorithms to measure driver fatigue. Anitha [16] proposed a novel two-fold yawning detection system based on an expert system. In the first part of the system, they used the face detection algorithm's skin tone detection and defined the boundaries of the face, then the blob dimensions for the mouth in the face containment are extracted. The system verified the yawning through a histogram of blobs is taken from the vertical projection of the lower face part. If the histograms values are satisfying with the threshold values, then yawning is confirmed. The proposed system achieved 94% performance for yawning detection. Kurylyak et al. [17] proposed an efficient approach to detect driver drowsy based on eye blinking. They used a web camera to acquire the driver image as input to the classifier using the Viola-Jones algorithm with Haar-like features to detect the driver's face and extract the eye region. A Kalman filter works with a set of discrete-time equations to compute and track the changes of the eye state. They calculate the frame's mutation for the thresholding value to detect the eyes' closure and opening. The frame processing algorithm is pointed out to distinguish the involuntary blinks from the voluntary ones. Experimental results of this proposed system presented 94% system accuracy to detect and determine the state of the eye. In another work by Jo et al. [18], researchers used the same method of Kurylyak et al. [17] to detect driver face, while they proposed a new eye drowsy detection method that combined two methods. Principal Component Analysis (PCA) is used to detect the eyes status in the daytime and Linear Discriminant Analysis (LDA) in the night. They applied Support Vector Machine (SVM) to classify the eye states to open and close through a specific interval of 3 minutes. Experimental results of this proposed system showed that 99% of the design work accurately to detect eyes drowsiness and driver distraction. However, the systems fail to recognize the eye in various high illuminations.

Abtahi et al. [19] proposed a new method of yawning detection based on the changes in the mouth geometric features and eye movement. They used color statistics for detecting skin color and texture. Therefore, they improved detection efficiency by using bounding rules for different color spaces (RGB, YCbCr, and HSV). They experimented on more than 500 images with varying reflections of light, skin color, haircuts, beards, and eyeglasses. Danisman et al. [20] proposed an automatic drowsy driver monitoring system to detect the eye blink duration. The proposed algorithm can catch the eye blinks movement in real-time using a webcam. Initially, they recognized the driver's face using the Viola-Jones algorithm, which is available in the OpenCV library of Python. Then, they discovered the positions of the pupils by used the symmetry property of the eye detector. The main drawbacks of that system are the presence of glasses and the various high illuminations, which affect the calculation and detection of the driver's drowsiness level. The proposed method achieved a 94% accuracy and a 1% false rate. Bergasa et al. [21] proposed a non-intrusive computer vision system for tracking driver's vigilance in real-time. The proposed method tried to test six parameters: face position and eye (eye rate, closure duration, blinks frequency, nodding frequency, and position of gaze). They used the fuzzy Logic approach to combine this feature and determined the driver drowsiness level. The system was an experiment in different driving environments (night and day) with other users. The system achieved 100% accuracy at night; however, the system did not work with glasses and bright days.

3. Proposed Approach

This study proposes a non-intrusive real-time drowsiness system based on the webcam video analysis. This section shows the detection algorithm used to detect the driver's drowsiness level by investigating and analyzing the different visual cues of the driver. The main two parameters are eye blink duration and mouth state information. Figure 1 presents a block diagram of the proposed monitoring system of the drowsy driver. The detection and monitoring approach consists of six stages as follows:

1. Image acquisition.
2. Face detection and tracking.
3. Eye-iris detection and tracking.
4. Mouth detection.
5. Mouth and eye information analysis.
6. Analysis driver state.

3.1. Image Acquisition

It provides images of a driver's face from the recorded video to observe and gather the visual cues and then determine the fatigue level. The MatlabR2016a environment provides an Image Acquisition toolbox that enables the user to connect to the scientific cameras. The proposed approach used a webcam tool that installs from support hardware properties in MATLAB to create a webcam object and snapshot function to acquire images in-stream video and convert them to the frame. Then, we manipulated the webcam object properties to be efficiently correlated with the HP laptop webcam.

3.2. Face Detection

Face detection is a computer technology that helps enhance human facial features in a digital image taken as input and used in various applications. This helps in processing the location and size of the human face and avoiding other objects [22]. There are many existing algorithms or methods for face detection technology, but the main difference is detection speed, accuracy, and purpose of use. Face detection algorithms work reasonably well with the detection of frontal and bright enough human faces images. It returns a sequence of analogous image coordinates where the human face is located and matched bit by bit. The proposal application assumes that the given input video detects only a single face (driver vehicle) in the camera view; otherwise, if more than one face, the system will detect the closest face to the center of the frame, as shown in Figure 2.

Implementing a face detection task is normalized to reduce and narrow down the domain of seeking the pupil and mouth detection. The eyes and mouth detector will not work if the desired face is not detected enough in the frame. However, pupil and mouth detection are located in the face area if the face is detected successfully by the face detector. After a comprehensive analysis, they suggest Viola and Jones's real-time detection technology. Viola and Jones have introduced this algorithm to assist us in detecting the desired facial feature image in a specific frame of the given video sequence of the vehicle driver. After getting the driver image and converted to a binary bit, as shown in Figure 3. The Viola-Jones algorithm passes the image through 4 steps to detect the driver's face objects.

Figure 1. proposed block diagram of the driver drowsy monitoring.

Step 1: Haar feature method:

Initial, the desired face scanned with the Haar feature method, which contains scalar values representing in two rectangles that can be horizontal or vertical in input image resolution 2424 pixels with a possible number of rectangles features 160,000.

Figure 3: (A) original image, (B) converted to gray image, (c) converted to binary image, (D) face detection.

Figure2: Flow diagram of Face Detector.

For example, the area where the eyes and mouth are located then passed this feature as an argument to the calculating [23]. Where I is an image, P is a pattern, $N \times N = \text{size}$.

$$\sum_{1 \leq i \leq N} \sum_{1 \leq j \leq N} I(i, j) 1_{p(i, j) \text{ is white}} - \sum_{1 \leq i \leq N} \sum_{1 \leq j \leq N} I(i, j) 1_{p(i, j) \text{ is black}} \quad (1)$$

Step 2: Integral image

The system calculates the Integral image and defines either the image contains the face or not at a very low computational cost using cumulative distribution functions.

$$II(i, j) := \begin{cases} \sum_{1 \leq s \leq i} \sum_{1 \leq t \leq j} I(s, t) & 1 \leq i \leq N \text{ end } 1 \leq j \leq N \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

$$\sum_{N_1 \leq i \leq N_2} \sum_{N_3 \leq j \leq N_4} I(i, j) = II(N_2, N_4) - II(N_2, N_3 - 1) - II(N_1 - 1, N_4) + II(N_1 - 1, N_3 - 1) \quad (3)$$

Step 3: Feature's selection Adaboost

This technique will remove all irrelevant features and combine only relevant features with their weight to evaluate and deciding either the image contains a face (1) or not (-1).

where, (X, Y) is a training example to the probability P , weights $w_i(1)$, $1 \leq i \leq n$ and $h_t = \text{decision stump}$. As the empirical loss goes to zero with T , so do both false positive $P(f^T(X) \neq 1 \mid Y = -1)$ and false negative rates $P(f^T(X) \neq -1 \mid Y = 1)$ as show in the equation 4.

$$\sum_{i=1}^n w_i(1) 1_{y_i \sum_{t=1}^T \alpha_t h_t(x_i) \leq 0} := P(f^T(X) \neq Y) \quad (4)$$

Step 4: Cascade method

This method increases processing power features by distributed every 10 features among single stages and sub-windows. Each sub-window will evaluate according to its feature by the cascading method. In the following picture show, the classifier triggers the evaluation and checks the characteristics of each sub-window. If the sub-window is classified as positive (face), it will be passing

through the steps. In the other case, the negative sub-window (not face) will immediately reject. This method will increase the performance power of the detected face by removing non-face-related windows at the beginning. Equation 5 defines the Cascade Decision rule to Obtain empirical loss.

$$f_{cascade}(X) = 2 \left(\prod_{p=1}^i 1_{f_{p,sp}^{Tp}(X)=1} - \frac{1}{2} \right) \quad (5)$$

where $f^{(p)}$ is a classifier with false positive.

Face tracking is handling by the Kalman filter. The Kalman filter method is an efficient way to estimates the position of a moving object based on its historical values in the next time frame. It can predict the state of a dynamic system from a sequence of uncertainty measurements by using a recursive adaptive filter. In addition, the Kalman filter was implemented to predict the dynamic change rate of the moving object and reduce the location error [24]. The following equation is used to track and predict the face position. Where x = target position, x_0 = initial position, v_0 = initial velocity, a = target acceleration and Δ is the time interval (3 second in this example) as computed in equation 6.

$$x = x_0 + v_0 \Delta t + \frac{1}{2} a \Delta t^2 \quad (6)$$

The face tracking method using a particle filter based on face position, face speed, and location error. However, this method fails when the brightness of the face, not enough due to background illumination and sudden move of the head. In Figure 4.a, 4.b, face tracking is trained to track the front face image and its maximum rotation between ± 15 to ± 50 degrees. However, in Figure 4.c, the detector fails when the rotation more than this and the alarm will rise.

Figure 4. Face tracking

3.3. Eye-Iris Detection and Tracking

The iris is a significant parameter that can be considered to assess the fatigue level of a driver. Tracking the eyelid and eye movement can reveal the size of the iris. Therefore, it is easy to determine the eye state through geometric calculations. As shown in Figure 4, in appropriate circumstances, the system first detects the driver's face and then segments the upper half of the face to identify the eyes. Once this region has been detected in a cropped image, it can help the system reduce the computing cost of the drowsiness levels. The cropped image is then converted into a binary image using the adaptive threshold technique to detect and replace all the white pixels in the input image with the value 1 and black pixels with the value 0. Following this, we use image processing, which provides morphological operations to process images based on shapes. The morphology technique only works with the relative ordering of pixel values; the value of each pixel in the picture is modified based on its neighboring pixel in the input image. To successfully perform a morphological operation, the size and shape of the neighborhood image need to be specified [25, 26].

Morphological image processing or calling a flat structuring element allows using functions such as segmentation, skeletonization, thinning, erosion, dilation, external boundary, and internal boundary. The system darkens the eye region by identifying the steel object to create a flat structuring element. This step is essential to eliminate false pixels and compute valid pixels. The eye detector focuses on the threshold and the rotation of the eye. Figure 5 illustrates a flat structuring element to detect the eyes. The iris is tracked successfully using a Kalman filter. This method uses a particle filter based on the iris' position, speed, and location error. However, this method fails when the brightness of the eye iris is not enough due to background illumination and sudden move of the head or camera resolutions. The eye size is different from one person to another; the system assumed at the beginning that; the

user is awake. Then, the threshold will be calculated and compared with the current eye ratio in the video Frame. The system cannot predicate eye iris positions correctly due to sudden movement of the head or camera resolutions.

Figure 5. a flat structuring element to detect the eye region

3.4. Mouth detection

Yawning is the most common sign of tiredness and drowsiness; hence, the system detects the driver's facial and eye movements. In the pre-processing stage, the system will crop the mouth region frame to determine the driver's state. Since the mouth is located in the lower part of the face, the mouth detectors will extract the mouth directly from the lower part of the face. After that, the system verifies mouth location using eye distance to detect the correct mouth segments. This is resolved by checking the boundaries of the mouth and eyes. Next, the system detects the mouth by calculating the connected object. The program will detect the upper and lower lips as two objects, assuming the mouth is open. When the mouth is closed, the program will calculate it as one object value; otherwise, the mouth would not be detected, and the program would return to zero value, as depicted in Figure 6. The monitored information will then pass to the fuzzy logic method to provide the driver's fatigue level.

Figure 6. Mouth detection.

3.5. Mouth and eye information analysis

After successfully detecting and tracking the facial features of the mouth and eyes, the system will compute the mouth and eye states by defining the two input parameters, which are the total number of black pixel areas and the Ratio of black pixels compared to the Ratio of white pixels.

Number of white pixels = sum (binary image)

Number of black pixels= sum (binary image) – number of white pixels

Ratio = number of black pixels/numbers of white pixels

The system uses the rules base to define the states of the eye and mouth. First, starting with the eye state, the system will compare the eye detection frame with the eye threshold to calculate the correct eye size ratio. Then, it will check if the Ratio of the left eye pixel is more than the threshold; the eye state is considered open or closed. If the eyes are detected to be closed for more than three seconds, an alarm will sound.

Similarly, the mouth state will be defined by checking the detected lips frame against the threshold to determine if the mouth is open or closed as shown in Figure 7.

Figure 7. Mouth processing.

3.6. Develop a fuzzy model for analysis driver behaviour

Fuzzy logic is a problem-solving algorithm that resembles human decision-making to provide a solution to a problem from vague or uncertain data. Fuzzy logic can describe fuzziness by representing a membership function and classifying the degree of truth of each element in a fuzzy set. A membership function is used to present a graphical representation of the fuzzy set based on the principle of the fuzzy rule IF-THEN [27]. Drowsiness is a type of fuzzy bodily state which is difficult to quantify objectively. Therefore, developing a fuzzy model can be an easy way to analyze driver behavior and determine their level of fatigue. In this system, the fuzzy inference system involves three main steps:

Step 1: Fuzzification

Define the two crisp input variables which are eye and mouth state with one output which is drowsy level as shown in Figure 8. Initially, we define the first input variable which eye state according to the ratio of eye closure as illustrate in below Table 1. Similarly, for mouth state, we define the input variable according to the ratio of open, half-open, close. A drowsy level is defined by three terms low, medium, and high. The Inputs include eyeState (open- half open- close) and mouthState (open- half open- close) values. The linguistic variables and terms for the inputs and output are given in below

Table 2. Furthermore, we use fuzzy logic to describe the fuzziness of the variables by represents the values from 0 to 1 using a set of input membership functions of both inputs and the degree of truth of each element in a fuzzy set is classified as presented in Figure 9.a, 9.b.

Table 1. eye closure

Ratio of eye closure	100-74%	73-35%	34-0%
Area of black pixels	small	medium	large
Drowsy level	low	medium	high

Table 2. linguistic variables and terms for the inputs and output.

Variable	Type	Fuzzy set	Membership function
Eye state	Input	Open, half open, close	Trapezoidal
Moth state	Input	Open, half open, close	Trapezoidal
Drowsy level	output	Low, medium, high	Trapezoidal

Step 2: Inference system

The if-then rules will be defined and evaluated in rule editor part of the Fuzzy Inference System. From the video frame sequence, the system can monitor the degree of the open/ close eye and mouth frame. Then, the system uses knowledge rules base fuzzy inference system and combines the two variables, which are eye and mouth state values. This process will help the system to classify

the feeling of the driver, whether it has low, medium, or high drowsy. [21] proposed fuzzy rules based on the ratio and threshold of open and close eyes. This is a set of rules for the knowledge base using IF-THEN logic defining in rule editor as presented in Table 3.

Figure 8. FIS editor showing the main simulation screen for driver drowsy

Step 3: De-Fuzzification

It converts the fuzzy input to the crisp value by uses the output membership functions to determine the drowsy output level. Using the rule viewer of MATLAB to interpret the entire fuzzy inference process at once and shows the diagram of membership functions influences the overall result. Figure 10 shows the rules' viewers of nine rows of plots rule and three columns of the inputs and output variable. The first two columns of plots yellow show the membership functions of the two input variables, and the last column represents the output membership function. The last plot in the last column shows the aggregate weighted decision for the given fuzzy rule system, which depends on the input values for the system.

(a)

(b)

Figure 9: FIS editor showing the MF (a): for eye state; (b) for mouth state.

Table 3. Rule editor

Rule Number	Rule
Rule 1	IF eye state is open OR mouth state is close THEN the drowsy level is low.
Rule 2	IF eye state is close OR mouth state is open THEN the drowsy level is high.
Rule 3	IF eye state is half open OR mouth state is half open THEN the drowsy level is high.
Rule 4	IF eye state is open AND mouth state is half open THEN the drowsy level is low.
Rule 5	IF eye state is half open AND mouth state is close THEN the drowsy level is medium.
Rule 6	IF eye state is open AND mouth state is open THEN the drowsy level is medium.

Figure 10. Rule's viewer.

Figure 11 presents the Surface Viewer of MATLAB three-dimensional curve of the entire fuzzy inference system. It is equipped with X (eyeState input), Y (mouthState input), and Z (drowsy level output) to allow the calculation time reasonable for complex problems. Also, the IF- then rules appear in 9 colors as displayed in the diagram of the fuzzy module; if the eyeState is closed at 0.3 degrees and the mouth state is opened at 0.9 degrees then, the drowsy level will be high.

Figure 11. Surface Viewer of fuzzy inference system.

4. Experimental Results

4.1. Offline Data analysis and training for eye and mouth

One of the principals aims of the project was to conduct a non-intrusive real-time drowsiness system based on the webcam video analysis. The system is focused on different visual cues of the driver to collect data and detect the driver's level of alertness. The systems focused on two parameters used to identify the driver's status: eye status and mouth status information. Firstly, we conducted an offline test on the MRL Eye Dataset [28] of the open and closed eye, as presented in Figure 12. Similarly, for mouth training, the OuluVS2 dataset contains different people's pictures with different ages, gender and is captured from different angles.

The main objective of using the dataset is to figure out the threshold of open and closed eyes and to test the system's capability for detecting the eyes state on different people or not. By testing different eyes samples, we figured out that if the eye image ratio exceeds the threshold, the eye will be considered open, otherwise closed. Also, the results shows that the program can detect all eyes in most pictures except the pictures with high reflection and lousy quality.

The system achieves promising results on detection eye status. Hence, it can conclude as results that the system can recognize the eye status successfully for both genders and wearing glasses or not. Besides, the system works effectively in case of offline detection due to the stability of the image. The error rate can be estimated due to the reflection and bad light of the pictures, see Table 4. However, the programs achieve 100% accuracy on offline detection mouth status. According to our results, the system can easily detect the mouth state from the pictures because no external obstacles affect how the system works, such as the sudden move, the distance between driver and camera, camera resolution, and background light.

Table 4. Offline MRL Eye Dataset analysis.

Subject	Gender	Glasses	Close	Open	No reflection	Low reflection	High reflection	Bad light	Good light	Accuracy of detection close eye	Accuracy of detection open eye
s0001	male	555	406	361	267	30	470	523	244	82%	80.7%
s0029	male	0	177	223	390	10	0	163	237	95.9%	96.8%
s0016	female	0	455	173	417	183	28	120	508	89.3%	90.1%
s0036	female	522	209	409	420	73	125	450	168	85%	84%

Figure 12. MRL Eye Dataset samples.

5. Conclusions and Future Work

This paper proposes a real-time drowsy detection system that monitors the eye and mouth of the driver through driving to measure his level of attention. The system was conducted as an offline experiment for eye and mouth states with different people and conditions. It achieved high accuracy compared to other works, about 87.875% for eye detection and 100% for mouth detection. Also, the system was tested in real-time with other videos recorded in a day and night, with different people wearing and not wearing glasses. The proposed method achieved 94.5% accuracy on real-time detection driver status. It also developed a fuzzy model for a mouth and eye variable input based on defined and evaluated if-then rules to investigate driver fatigue levels. Finally, it visualizes the eye state using a histogram chart to check and examine the difference between open and closed eyes.

Driver drowsy is a major cause of road accidents and economic risk. Using a Webcam tool to detect fatigue is still not efficient enough because the face cannot be detected when the driver moves their head quickly or suddenly. Also, facial features cannot discover in the video frame under various lighting conditions or wearing sunglasses, which gives inaccurate data in the fatigue warning system. Hence, in the future, the system should use an Infrared Cameras-sensitive camera to allow the system works robustly in any lighting conditions. The proposed approach cannot detect the driver's mouth because a typically yawning gesture covers the mouth while yawning. Hence, in the future, the system should add another efficient parameter which is steering wheel angles based on using deep learning and Recurrent neural network [29, 30]. The steering wheel angles sensor is widely used to measure the driver behavior and determine the fatigue level in real-time. The system should improve the response time to get accurate results and avoid any wrong alarms of the drowsy detection system. Cloud computing techniques can help reduce the incompatibility of data and the cost of using fast hardware and software [31, 32].

References

1. National Safety Council (NSC). Online source. [Accessed on 5.10.2021]. <https://www.nsc.org/road/safety-topics/fatigued-driver>
2. Salvati L, d'Amore M, Fiorentino A, Pellegrino A, Sena P, Villecco F. On-road detection of driver fatigue and drowsiness during medium-distance journeys. *Entropy*. 2021 Feb;23(2):135.
3. Kaplan, S.; Guvensan, M.A.; Yavuz, A.G.; Karalurt, Y. Driver behavior analysis for safe driving: A survey. *IEEE Transactions on Intelligent Transportation Systems* 2015, 16, 3017–3032.
4. AL-Balushi AI, Yousif J, Al-Shezawi M. Car accident notification based on Mobile cloud computing. *International Journal of Computation and Applied Sciences IJOCAAS*, Volume2. 2017 Apr(2).
5. Mehta, S.; Dadhich, S.; Gumber, S.; Jadhav Bhatt, A. Real-time driver drowsiness detection system using eye aspect ratio and eye closure ratio. *Proceedings of international conference on sustainable computing in science, technology and management (SUSCOM)*, Amity University Rajasthan, Jaipur-India, 2019.
6. Madhan Kumar, A.; Kabilan, M.; Karthikeyan, S.; Sathiyapriya, S. Drowsy driving warning and traffic collision information system using IoT 2019.
7. Herrera-Granda, E.P.; Caraguay-Procel, J.A.; Granda-Gudiño, P.D.; Herrera-Granda, I.D.; Lorente-Leyva, L.L.; Peluffo-Ordóñez, D.H.; Revelo-Fuelagán, J. Drowsiness Detection in Drivers Through Real-Time Image Processing of the Human Eye. *Asian Conference on Intelligent Information and Database Systems*. Springer, 2019, pp. 626–637.
8. Roig, A.B.; Morales, M.; Espinosa, J.; Perez, J.; Mas, D.; Illueca, C. Pupil detection and tracking for analysis of fixational eye micromovements. *Optik* 2012, 123, 11–15.
9. Shelke, P.; Magar, S.; Jawkar, R. Real Time Detection System of Driver Fatigue 2016.
10. Azim, T.; Jaffar, M.A.; Mirza, A.M. Automatic fatigue detection of drivers through pupil detection and yawning analysis. 2009 fourth international conference on innovative computing, information and control (ICICIC). IEEE, 2009, pp. 441–445.
11. Liu, W.; Sun, H.; Shen, W. Driver fatigue detection through pupil detection and yawing analysis. 2010 International Conference on Bioinformatics and Biomedical Technology. IEEE, 2010, pp. 404–407.
12. Yousif JH, Saini DK. Cloud computing and accident handling systems. *International Journal of Computer Applications*. 2013 Jan 1;63(19).
13. Saini DK, Yousif JH. Reduction in accidental death: determining driver behavior using fuzzy theory. In *Annual International Commerce Conference 2016* (pp. 1-7).
14. Azim, T.; Jaffar, M.A.; Mirza, A.M. Fully automated real time fatigue detection of drivers through fuzzy expert systems. *Applied Soft Computing* 2014, 18, 25–38.
15. van den Berg, J.; Neely, G.; Wiklund, U.; Landström, U. Heart rate variability during sedentary work and sleep in normal and sleep-deprived states. *Clinical physiology and functional imaging* 2005, 25, 51–57.
16. Anitha, C.; Venkatesha, M.; Adiga, B.S. A two fold expert system for yawning detection. *Procedia computer science* 2016, 92, 63–71.
17. Kurylyak, Y.; Lamonaca, F.; Mirabelli, G. Detection of the eye blinks for human's fatigue monitoring. 2012 IEEE International Symposium on Medical Measurements and Applications Proceedings. IEEE, 2012, pp. 1–4.
18. Jo, J.; Lee, S.J.; Kim, J.; Jung, H.G.; Park, K.R. Vision-based method for detecting driver drowsiness and distraction in driver monitoring system. *Optical Engineering* 2011, 50, 127202.
19. Abtahi, S.; Hariri, B.; Shirmohammadi, S. Driver drowsiness monitoring based on yawning detection. 2011 IEEE International Instrumentation and Measurement Technology Conference. IEEE, 2011, pp. 1–4.
20. Danisman, T.; Bilasco, I.M.; Djeraba, C.; Ihaddadene, N. Drowsy driver detection system using eye blink patterns. 2010 International Conference on Machine and Web Intelligence. IEEE, 2010, pp. 230–233.
21. Bergasa, L.M.; Nuevo, J.; Sotelo, M.A.; Barea, R.; Lopez, M.E. Real-time system for monitoring driver vigilance. *IEEE Transactions on Intelligent Transportation Systems* 2006, 7, 63–77.
22. Boda, R.; Priyadarsini, M.J.P.; Pemeena, J. Face detection and tracking using KLT and Viola Jones. *ARPN Journal of Engineering and Applied Sciences* 2016, 11, 13472–13476.
23. 20Gejguš, P.; Šperka, M. Face tracking for expressions simulations. *International Conference on Computer Systems and Technologies*. Citeseer, 2003.
24. Komogortsev, O.V.; Khan, J.I. Kalman filtering in the design of eye-gaze-guided computer interfaces. *International Conference on Human-Computer Interaction*. Springer, 2007, pp. 679–689.
25. Al-Hatmi MO, Yousif JH. A review of Image Enhancement Systems and a case study of Salt & pepper noise removing. *International Journal of Computation and Applied Sciences (IJOCAAS)*. 2017;2(3):171-6.
26. Hasoon FN, Yousif JH, Hasson NN, Ramli AR. Image enhancement using nonlinear filtering based neural network. *Journal of Computing*. 2011 May;3(5):171-6.
27. Yager, R.R. Expert systems using fuzzy logic. In *An introduction to fuzzy logic applications in intelligent systems*; Springer, 1992; pp. 27–44.
28. Rahman, M.M.; Islam, M.S.; Jannat, M.K.A.; Rahman, M.H.; Arifuzzaman, M.; Sassi, R.; Aktaruzzaman, M. EyeNet: An Improved Eye States Classification System using Convolutional Neural Network. 2020 22nd International Conference on Advanced Communication Technology (ICACT). IEEE, 2020, pp. 84–90.
29. Abtahi, S.; Omidyeganeh, M.; Shirmohammadi, S.; Hariri, B. YawDD: A yawning detection dataset. *Proceedings of the 5th AC multimedia systems conference*, 2014, pp. 24–28.
30. Shahverdy M, Fathy M, Berangi R, Sabokrou M. Driver behavior detection and classification using deep convolutional neural networks. *Expert Systems with Applications*. 2020 Jul 1;149:113240.
31. Yousif JH, Sembok T. Recurrent neural approach based Arabic part-of-speech tagging. In *proceedings of International Conference on Computer and Communication Engineering (ICCCE'06) 2006* (Vol. 2, pp. 9-11).
32. Škrjanc I, Andonovski G, Ledezma A, Sipele O, Iglesias JA, Sanchis A. Evolving cloud-based system for the recognition of drivers' actions. *Expert Systems with Applications*. 2018 Jun 1;99:231-8.
33. Yousif JH, Alattar NN. Cloud management system based air quality. *International Journal of Computation and Applied Sciences (IJOCAAS)*. 2017;2(2).