Drowsiness Detection and Rest Stop Suggestion

Sabari Mathavan R B¹, Rohitram V¹, Ashhwath C¹, Sasikumar P¹,*

¹ School of Electronics, Vellore Institute of Technology, Vellore, India
* sasikumar.p@vitstudent.ac.in

Abstract — In recent years, driving accidents have been increasing gradually, with most of them due to drowsiness. To tackle this problem, various methods and systems were developed to alert the driver behind the wheels, and to prevent accidents. A few approaches include heartbeat monitoring, pulse rate check, blink, and yawn detector. This paper discusses a driver mechanism, which is capable of detecting fatigue in real-time, using the Eye Aspect Ratio – EAR parameter, and also provides a database with an interface for managers to track drivers. When the driver is tired, an email will also be sent to the manager. On the drivers’ side, there will be a buzzer noise to wake the driver up and nearby resting locations will be shown to the driver. Further, mapping to the nearby resting location of the driver’s choice is shown, if the driver feels too drowsy and feels the need for rest. It is essential to fabricate and construct this complete system via IoT and OpenCV methods to address this issue.

Keywords — Drowsiness detection; Computer vision; API call; Eye Aspect Ratio; Facial Landmarks

1. Introduction

With the rising economy in the current fast-paced world, the strain on long-distance monotonous rides has increased dramatically. This is solely due to the extremely short delivery completion deadlines. Drivers are forced to travel great miles in a short amount of time to make money, which can be hazardous as it leads to accidents. These are mostly the result of human mistakes like reckless driving, driving under influence (like alcohol, drugs, etc.), and fatigued and drowsy drivers, or "driver fatigue.” These result in traffic accidents, causing problems for fellow travelers as well as to those who require transportation of products. Fatigue from drivers has been the main concern for innumerable mishaps which is caused by drowsiness, bad road and weather condition. Drowsiness is a state of reduced attentiveness combined with a proclivity to fall asleep. When it happens while driving, it usually results in a serious collision. Driver fatigue is a major cause of road traffic accidents, according to figures from the Ministry of Transport of the Government of India (GoI). Studies show that in India, which has a relatively weak economy and poor infrastructure facilities, the drivers are forced to have hectic drive schedules with high work pressure for financial or reward-based earnings [1]. In the United States of America, fatigue-related accidents account for 2.2% to 2.6% of all fatal crashes every year [2]. But according to AAA Foundation for Traffic Safety, studies show that the statistics may go up a little further. Drowsiness was found in 8.8% to 9.5% of all collisions investigated, and 10.6% to 10.8% of accidents that resulted in serious property damage, airbag deployment, or injury, according to the research. [3]. Many such crashes occur when the driver is tired of the journey and falls asleep or loses control of the vehicle. William Van Tassel, manager of Driver Training for AAA, says, “Don’t be fooled, the only antidote for drowsiness is sleep” [4]. Sleep is a prime factor for the normal functioning of the body.
To prevent these situations the presence of a drowsiness detection system seems vital. The Driver Drowsiness Detection System (DDDS) can be classified into 2 basic measures – Physiological measure, Behavioural measure. Physiological measures require electrodes, which are placed in contact with the body, during the journey. The physiological indicators of the driver vary as a result of weariness, and these changes can be examined to evaluate the level of drowsiness. A few indicators include - Electroencephalogram (EEG), Electrocardiogram (ECG), Electrooculogram (EoG). These physiological indicators predict drowsiness more accurately than the behavioural measures, but the requirement of electrodes may cause disturbance to the driver while driving. The behavioural measures make use of observable features like the state of the eye, eye blinking rate, head movement and yawning. Though these systems may underperform during low-lit conditions and also depends on the road conditions, etc., these do not cause discomfort to the driver, which makes them advantageous over the physiological measures. Various methods were proposed and implemented based on the behavioral approach, for the betterment of detection of drowsy state. Md. Yousuf Hossain et al. (2018) presented a paper on “IoT based Real-time Drowsy Driving Detection System for the Prevention of Road Accidents” [5]. A real-time drowsiness detection system that detects the drowsiness of a driver based on the Eye Aspect Ratio (EAR) is created and the state of drowsiness is identified when the EAR value of a person, reduces below 0.25. Alerting the driver through a buzzer is implied by them along with electronic mail being sent to the owner. Anil Kumar Biswal et al. (2021) addressed a process that is similar to the previous method [6]. Only it uses a robust alert system for eye blinking detection using Eye Aspect Ratio (EAR) and Euclidean distance of the eye. Video Stream Processing is used to detect the blinking of the eye. In this work, the highlight is that the system is not external but is inbuilt on intelligent building vehicles that can automatically avoid a drowsy driver. 

Tereza Soukupova et al. (2016) proposed a method of eye detection using Eye Aspect Ratio (EAR) to check the eye-opening in each frame [7]. SVM classifier is also used to detect patterns of EAR values in a short temporal window. Despite, to account for less timing of blinking eyes, a classifier that has a larger temporal window is trained in the required way. Anna Liza A. Ramos et al. (2019) developed API Based Histogram of Oriented Gradient and support vector machine for drowsiness detection [8]. This paper works by analyzing the regression range, eye closure sensitivity with a random forest algorithm and predicted whether the person is drowsy or not while driving. Their model detected drowsy activity with 91.67% in a 0° camera angle and 93.33% accuracy in both +45° and -45° camera angles. Sang-Ho Jo, et al. (2018) made studies on heart rate change during drowsy driving [9]. Six healthy volunteers were used to check their mean heart rates (HR) during morning and night. The mean heart rates were found to be increasing during daytime and vice versa. The biggest advantage of this work is that the drivers are monitored 24/7, such that suitable precautions can be taken. Seok-Woo Jang et al. (2020) presented a system to reduce major accidents due to drowsiness [10]. In this proposal, machine learning models along with IoT for clear image processing are used and thereby reducing the risk of not capturing drowsiness symptoms. In addition, a voice recognition application program interface with speech to text (Speech Signal Processing) is also included for engaging people while driving with their favorite songs, current news and thus, making them active.

In this proposed system, a mechanism is implemented to detect and alert the driver and the manager on the condition of the driver fatigue and location via IoT and OpenCV. The driver is monitored by a camera, and the facial landmarks are detected using the Haar Cascade classifier. These properties assist to evaluate driver weariness and immediately alert him using a buzzer and send the driver's current location (for tracking purposes) as an email to the manager of the vehicle. As for the drivers, a list of nearby rest stops is presented if drowsiness/fatigue is identified so the driver can choose from the list and head to the chosen stop, then rest for a while and continue the journey, if required. An app named Diver's Lounge is also designed to track drivers and their state by the managers of the company. The information for the app is initially saved in a database, upon fatigue detection.
2. Proposed System

The objective of this paper is to create a fully automatic drowsiness detection system for handling the drowsy driver detection problem. To make the driver awake when he faces fatigue, an alarm/buzzer is sounded to alert the driver. Further, if the driver feels still drowsy and wants to take a rest at a lodging place, the nearest lodging places are shown using Google Places API, and the driver is asked for a choice of place. Once the driver chooses his choice of resting place, a map is drawn between the user's location and the chosen place, and details about the destination are provided. This helps the driver to navigate to the nearest resting location of his choice.

To detect the person's state of drowsiness the Eye Aspect Ratio (EAR) parameter is used. The EAR parameter is the ratio of distances between the vertical eye landmarks and the distances between the horizontal eye landmarks. The conventional threshold value of EAR for a state of awakeness is 0.25. If a person feels drowsy/fatigue, the eyes tend to close thus decreasing the value of EAR. With 0.25 as the boundary between the states of being awake or drowsy, if the EAR drops lower than 0.25, the person is concluded drowsy. But even a blink of an eye has an EAR value less than 0.25. To overcome this drawback of false drowsiness during a blink, successive frames are noted. If the EAR value remains lower than 0.25 for some continuous finite number of frames then the state of drowsiness is concluded. This finite number is dependent on the frame rate of the camera. Also, the manager/company is notified through electronic mail. To keep track of the location, time, and driver information, a database is created to store the values. Google Firestore was used, as the database system. Fig 1. shows the flowchart of the proposed system.

3. Technique

3.1. Drowsy detection using OpenCV

The drowsiness of a person can be detected using various parameters, e.g., closing eyes, yawn, pulse rate, etc. In this paper, to test the user's drowsiness level, the EAR parameter is used. A webcam is used to capture the live movements of the eyes, along with the "OpenCV" library.

![Flowchart of the proposed system.](image-url)
Each frame of the video is captured and is then converted to the grayscale format. This is because the grey channel is computationally cheap and uses less computational power because it just has one black-and-white channel. The Haar Cascade Classifier is used to detect facial landmarks. Facial landmarks are the fundamental elements that make up a person's face, and these include the eyes, eyebrows, nose, mouth, and jawline. The Haar Cascade Classifier is based on a machine learning approach that involves training a cascade function from a large number of positive and negative images. It is then used to detect the objects in the other images based on the training. In comparison to other object detection techniques, Haar – based classifiers involve less computations and thus reaching higher execution speeds. Moreover, relatively small dataset is enough to train the classifier, as there is no need to train the Haar features. The Haar Cascade classifier's output is also accurate, making it a popular method for object detection.

To detect the closing of eyes, the EAR parameter, which stands for Eye Aspect Ratio, as explained above is used. The Euclidean Distance of the Eye Region is utilised to compute the distances. (1)

\[ D(P, Q) = \left( \sum_{i=1}^{n} (Q_i - P_i)^2 \right)^{1/2} \]  

The code uses the “dlib” library to give the code ability to put up landmarks on the image and track it. It can detect and track a face and its features. As explained above there are a total of 68 landmarks out of which 12 landmarks denote eyes (6 for each eye). Using these points, the Euclidean distance can be calculated and the EAR is obtained, which shows the size of the shape obtained by joining the eye landmarks. A frame is classified as a closed eye frame if the EAR of a frame is lower than 0.25.

**Figure 2. (a).** Six landmarks of the eye while eyelids are open.  
**Figure 2. (b).** Six landmarks of the eye while eyelids are closed. [11]

The EAR value is computed using,

\[ EAR = \frac{\|p_2 - p_6\| + \|p_3 - p_5\|}{2\|p_1 - p_4\|} \]  

The 2D landmark locations are \(p_1, p_2, p_3, p_4, p_5,\) and \(p_6,\) as shown in Fig. 2. Since both eyes blink simultaneously, the EAR of both eyes is averaged as in (2). Not all closed-eye frames represent drowsiness. This is because a blink is also classified as a closed eye frame. So, a threshold value is given to counting the number of continuously closed eye frames. If the count exceeds the threshold value, a conclusion is drawn stating that the driver is feeling drowsy. The threshold value is determined based on the frame rate of the camera device that's being used. According to Healthline, a renowned health information provider website, the typical duration of a single-eye blink usually varies between 100 and 400 milliseconds for each individual [12]. For safer reasons, a window of 800ms is took. The threshold value for 800ms corresponds to 48 frames, in this case, since the
frame rate of the camera used is 60 frames per second. If the eye closing is detected continuously for 48 frames, a conclusion is drawn that the driver is drowsy. Once the state of drowsiness is concluded, an alarm/buzzer is used to alert the driver from drowsiness/fatigue. The “playsound” python library is used to produce the sound. Further, the control is transferred to choose the nearest resting place, to intimate the manager/organization of the driver through electronic mail, and also to append the information to the firebase.

3.2. Mapping the locational coordinates
The locational coordinates of a place are always represented as a set of 2 values - latitude, longitude. These 2-dimensional coordinate values can represent any place on earth. The longitudes range from -180 to +180 from the prime meridian. The latitudes range from -90 to +90, with 0 as the equator. These values are represented in the form of degrees, minutes, and seconds. For simplicity, the data is converted into decimal format. To determine the locational coordinates from the system, Windows runtime API is used. It provides the results within a distance error of 10m if GPS is being used, or up to 500m if Wi-Fi is being used for tracking. The Windows API along with its Geolocator function provides the results in the form of geographic locational coordinates (decimal).

To collect information about the nearby resting spots, or restaurants for resting/refreshments, Places API of Google Developers is used. The Places API is a service that returns information about places using HTTP requests [13]. The locational coordinates obtained from the previously mentioned steps are passed as parameters to the API call, along with the place type attribute, which is lodging. The search results are returned in the form of JSON text. The obtained text sets are further processed to get the required parameters - location name, coordinates. The obtained locations along with the distance of the place from the current location of the user are displayed and the user is asked to make a decision on the place he/she wants to visit or rest. Once the user selects his/her choice of resting location, a map is configured with the source (user’s location) and destination with marker, using gmaps. “gmaps” is a Jupyter plugin for embedding Google maps in Jupyter notebooks. It is designed to help visualize and interact with geographical data [14]. Routing between the source and the destination is done for better visualization. The final map is created after the addition of the direction layers. The map is displayed to navigate to the destination. Along with the map, the distance of the place from the current location of the user, the approximate time that is taken to reach the destination, and the address of the destination is also provided.

3.3. Cloud Firestore and Email
Storing data is important for future uses. For example, to have a track of tiredness record of each driver. So, to analyse the records of each driver, values such as the driver’s name, vehicular registration number, timestamp, and locational coordinates of the place where tiredness was detected, are stored. To store these data values, Google Cloud Firestore was used. Cloud Firestore is a worldwide NoSQL document database that makes it simple to store, sync, and query data for your mobile and online apps. Google Cloud Firestore is configured with the python file using serviceAccountKey, which is a JSON file provided by Firestore to access a particular database account. Data is sent over to the Firestore, and recorded under the database table named “records”.

For further notifications, the email sending option to the manager. Is enabled the email is sent using the SMTP protocol. The Simple Mail Transfer Protocol (SMTP) is an internet standard communication protocol for electronic mail transmission. Python uses the native “smtplib” library to send emails from a specified user to another user. The smtplib module creates an SMTP client session object that can be used to send email to any system on the Internet that has an SMTP or ESMTP listener daemon. Values like the vehicular registration number, the driver’s name, the locational coordinates, and the time of fatigue detection are sent in the mail as the message.
3.4. Driver’s Lounge

With the backends finished, a user interface is needed to show the drivers the various places that are recommended to be stopped in. Drivers Lounge (an app) is used to show the details of the drivers. This is done by getting the data from Cloud Firestore that was stored in the previous model, and using that to display driver details such as the driver’s name, plate number, and when the data was received. The location is also displayed on the map. This is done using Flutter as it provides a limitless UI with ease, as well as being cross-platform.

4. Implementation and results

As proposed the system could detect the eye landmarks perfectly and can classify the state of the driver based on EAR values. A few tech stacks and gears were used to get things done. The proposed idea is implemented using python 3.6.0, which provides us with OpenCV, and dlib libraries to work on with the video capturing and facial landmark recognizing parts. To capture the frame in-built webcam was used, HP TrueVision HD camera. This provides us with a 720p clarity video at 60 fps. For API calling of Places API and Maps API, Google cloud services were used, and similarly, to store and retrieve values Google Cloud Firestore was utilized and applications were developed using Google Flutter.

![Figure 3. (a) The EAR value when eyes are open](image)

![Figure 3. (b) The EAR value when eyes are closed](image)

From fig. 3. (a) it can be noticed that the EAR value lies well above 0.25 when the eyes are open. As the eyes close, the EAR value drops. This is shown in fig. 3. (b).

![Figure 4. (a) Email received at the receiver’s mail address](image)
The system throws an alert message whenever the EAR value goes below 0.25. When the EAR falls below 0.25 continuously for 48 frames, the alarm is sounded and a rest stop suggestion is provided. Also, the data like vehicle number, driver name, coordinates, time are stored over Firestore, fig 4 (a), and similarly a mail containing these data is sent to the manager, fig 4 (b). The rest stop suggestion works on the Google Places API. The locational data is passed on through the API call and the nearest resting places are suggested to the driver, to select a place as shown in fig. 5.

Once the driver selects the place he wishes to go, a mapping is done between the driver’s location and the destination chosen, as shown in fig 6 (a).

Figure 4. (b) Screenshot of the Firebase data system

Figure 5. Drivers are asked to select their choice of resting place.

Figure 6. (a) Mapping between the source (marker A) and destination (marker B)

Figure 6. (b) Additional information for assistance
Additional information like distance of the destination from the source, average time to reach there, and the address of the destination are provided to the driver to assist him to arrive at the destination. The records in the database are shown to the manager in the form of an app. This app is provided to the manager/owner of the vehicle for a regular time to time update.

**Figure.** 7 (a) shows the list of all drivers that are available in the database whereas fig 7 (b) shows the location of a particular driver, where the user was detected drowsy last.

5. Conclusion

The main object of this research paper is to build a real-time drowsiness detection system that could detect the drowsiness of a driver/person and suggest nearby stop locations in case the person feels weak. Most of the current drowsiness detection systems use heart rate as a parameter for detection. The driving heart rate increments from $85 \pm 5.6$ beats/min, the day time mean heart rate, to $89.8 \pm 5.6$ beats/min. But when the driver is detected drowsy, it decreases to $81.5 \pm 9.2$ beats/min, as per [9]. These changes in heart rate may not be so confident in detecting drowsiness while driving. On the other hand, the proposed detection system uses the EAR value to determine the state of drowsiness. Using webcams and external cameras, the facial points of the face are monitored constantly and the EAR values get updated. If the EAR value drops below a specified threshold (0.25) for a certain number of frames (48 frames), an alarm is sounded, and a navigation option to a nearby lodging location is provided. If the driver is detected drowsy too often then electronic mail is sent to the manager/owner. This system may prove worth to long-distance drivers, especially for truck/cab drivers. Further, features like dynamic mapping, voice integration, can be added to make the system better and interactive.

References

[1] Mahajan, Kirti & Velaga, Nagendra & Kumar, Akhilesh & Choudhary, Pushpa. (2019). Effects of driver sleepiness and fatigue on violations among truck drivers in India. International Journal of Injury Control and Safety Promotion. 26. 1-11. 10.1080/17457300.2019.1660375.
[2] McArthur, Adam & Kay, Jonathan & Savolainen, Peter & Gates, Timothy. (2013). Effects of Public Rest Areas on Fatigue-Related Crashes. Transportation Research Record: Journal of the Transportation Research Board. 2386. 16-25. 10.3141/2386-03

[3] AAA Foundation for Traffic Safety. (2018, February). Prevalence of Drowsy Driving Crashes: Estimates from a Large-Scale Naturalistic Driving Study. https://aaafoundation.org/prevalence-drowsy-driving-crashes-estimates-large-scale-naturalistic-driving-study/

[4] AAA Foundation for Traffic Safety. (2018, February 2). Drowsy Driving: Don’t Be Asleep at the Wheel. https://newsroom.aaa.com/2018/02/drowsy-driving-dont-asleep-wheel/

[5] M. Y. Hossain and F. P. George, 2018, IOT Based Real-Time Drowsy Driving Detection System for the Prevention of Road Accidents, International Conference on Intelligent Informatics and Biomedical Sciences (ICIIBMS), 2018, pp. 190-195, doi: 10.1109/ICIIBMS.2018.8550026.

[6] Biswal, Anil Kumar & Singh, Debabrata & Pattanayak, Binod & Samanta, Debabrata & Yang, Ming-Hour. (2021). IoT-Based Smart Alert System for Drowsy Driver Detection. Wireless Communications and Mobile Computing. 2021. 1-13. 10.1155/2021/6627217.

[7] Soukupová, T., & Cech, J. (2016). Real-Time Eye Blink Detection using Facial Landmarks. In 21st computer vision winter workshop, Rimske Toplice, Slovenia;

[8] Anna Liza A. Ramos, Jomar Christian Erandio, Dean Hur T. Mangilaya, Nathaniel Del Carmen, Erika May Enteria, Leonilo Jean Enriquez, “Driver Drowsiness Detection Based on Eye Movement and Yawning Using Facial Landmark Analysis”, International Journal of Simulation--Systems, Science & Technology, 2019.

[9] Jo, S. H., Kim, J. M., & Kim, D. K. (2019). Heart Rate Change While Drowsy Driving. *Journal of Korean medical science*, **34**(8), e56. [https://doi.org/10.3346/jkms.2019.34.e56](https://doi.org/10.3346/jkms.2019.34.e56)

[10] Jang, Seok-Woo & Ahn, Byeongtae. (2020). Implementation of Detection System for Drowsy Driving Prevention Using Image Recognition and IoT. Sustainability. 12. 3037. 10.3390/su12073037.

[11] Dhruv Pandey, Medium. (2021, April). Eye Aspect Ratio (EAR) and Drowsiness detector using dlib. [https://medium.com/analytics-vidhya/eye-aspect-ratio-ear-and-drowsiness-detector-using-dlib-a0b2c292d706](https://medium.com/analytics-vidhya/eye-aspect-ratio-ear-and-drowsiness-detector-using-dlib-a0b2c292d706)

[12] Healthline Media, Inc. (2020, September 24). How Many Times Do You Blink in a Day? [https://www.healthline.com/health/how-many-times-do-you-blink-a-day](https://www.healthline.com/health/how-many-times-do-you-blink-a-day)

[13] Google. (n.d.). Google. [https://developers.google.com/maps/documentation/places/web-service/overview](https://developers.google.com/maps/documentation/places/web-service/overview).

[14] API documentation. API documentation - gmaps 0.8.3-dev documentation. (n.d.). [https://jupyter-gmaps.readthedocs.io/en/latest/api.html](https://jupyter-gmaps.readthedocs.io/en/latest/api.html)

[15] R. Seetharaman, S. Sriradha and S. Mootha, "Detection and State Analysis of Drowsiness using Multitask Learning with Neural Networks," 2020 Fourth International Conference On Intelligent Computing in Data Sciences (ICDS), 2020, pp. 1-8, doi: 10.1109/ICDS50568.2020.9268740.

[16] Sathish Kumar. R; Nivedha. K; Anitha. K; Jayaprakash. D. "IOT Based Driver Drowsiness Detection and Health Monitoring System". *European Journal of Molecular & Clinical Medicine*, **7**, 9, 2020, 1581-1590. doi: 10.31838/ejcm.07.09.173

[17] Culjak, I., David Abram, T. Pribanić, H. Džapo and M. Cifrek. “A brief introduction to OpenCV.” *2012 Proceedings of the 35th International Convention MIPRO* (2012): 1725-1730.

[18] A. Mittal, K. Kumar, S. Dhamija and M. Kaur, "Head movement-based driver drowsiness detection: A review of state-of-art techniques," 2016 IEEE International Conference on Engineering and Technology (ICETECH), 2016, pp. 903-908, doi: 10.1109/ICETECH.2016.7569378.

[19] Mehta, Sukrit and Dadhich, Sharad and Gumber, Sahil and Jadhav Bhatt, Arpita, Real-Time Driver Drowsiness Detection System Using Eye Aspect Ratio and Eye Closure Ratio (March 20, 2019). Proceedings of International Conference on Sustainable Computing in Science,
Technology and Management (SUSCOM), Amity University Rajasthan, Jaipur - India, February 26-28, 2019

[20] Hoyoung, Lee, et al. “Drowsiness Detection with OpenCV.” *PyImageSearch*, 17 Apr. 2021, www.pyimagesearch.com/2017/05/08/drowsiness-detection-opencv/

[21] Juvale H.B. et al. (2009) Design of DroDeASys (Drowsy Detection and Alarming System). In: Ao SL., Rieger B., Chen SS. (eds) Advances in Computational Algorithms and Data Analysis. Lecture Notes in Electrical Engineering, vol 14. Springer, Dordrecht.

[22] Jeong C, Kim T. Eye blink detection using algorithm based on dlib and opencv library for game players in competitive environments. Jirmeps [internet]. 19jun.2021 [cited 30aug.2021];16(2):33-45.

[23] Satman, M.H., & Altunbey, M. (2014). Selecting Location of Retail Stores Using Artificial Neural Networks and Google Places API. *International Journal of Statistics and Probability*, 3, 67.

[24] Chatterjee, Nilanjan & Chakraborty, Souvik & Decosta, Aakash & Nath, Asoke. (2018). Real-time Communication Application Based on Android Using Google Firebase. IJARCSMS.

[25] Kuzmin N., Ignatiev K., Grafov D. (2020) Experience of Developing a Mobile Application Using Flutter. In: Kim K., Kim HY. (eds) Information Science and Applications. Lecture Notes in Electrical Engineering, vol 621. Springer, Singapore. https://doi.org/10.1007/978-981-15-1465-4_56

[26] Jackson, Paul & Hilditch, Cassie & Holmes, Alex & Reed, Nick & Merat, Natasha & Smith, L. (2011). Fatigue and road safety: a critical analysis of recent evidences.

[27] Paul E. Black, "Euclidean distance", in *Dictionary of Algorithms and Data Structures* [online], Paul E. Black, ed. 17 December 2004. Available from: https://www.nist.gov/dads/HTML/euclndstnc.html

[28] Soukupová, T., & Cech, J. (2016). Real-Time Eye Blink Detection using Facial Landmarks. In 21st computer vision winter workshop, Rimske Toplice, Slovenia.