ISSN 1533-9211
DOI 10.5281/zenodo.7273935

DEVELOPMENT OF DROWSINESS DETECTION TECHNOLOGY BASED ON COMMAND LINE INTERFACE APPLICATION

EUIS NINA SAPARINA YULIANI[1], I PUTU GEDE ADIATMIKA[2], KETUT TIRTAYASA[3] and HARDIANTO IRIDIASTADI[4]

[1]Industrial Engineering Study Program, Mercu Buana University. Email: nina.yuliani@mercubuana.ac. id
[2, 3] Ergonomics -Work Physiology, Medical Faculty, Udayana University.
[4] Industrial Engineering Study Program, Industrial Technology Faculty, Institut Teknologi Bandung.

Abstract

The high number of accidents in Indonesia, of which is caused by drivers who experience fatigue and drowsiness while driving. Drowsiness and fatigue can be caused by the number of activities carried out, lack of sleep, long duration of driving, and an unfit body condition. Various indicators that are objective and subjective to assess the level of drowsiness and fatigue have been developed. However, objective indicators that can detect driver fatigue through sleep levels in real-time have not been widely developed, especially in Indonesia. Therefore, this study will develop drowsiness detection technology based on an intelligent system application, using three parameters, namely eye blinking detection, yawning mouth duration (yawn detection), and head pose detection, which is designed to detect driver drowsiness by receiving input in the form of image capture from the face and providing an output in the form of a warning signal. Facial area detection uses the facial landmark method that works by allocating points on the face to determine the biological shape of the human face. In processing facial images, Dlib facial landmarks and Open CV will be used, with the Python programming language. This research will be divided into several stages of technology development. In the first stage, a command line interface (CLI) application will be developed to simplify the development process. The results of the first stage of research are technology based on the Command Line Interface Application. The accuracy test of this application uses a video dataset of four drivers, where the test results state that the CLI application has an accuracy rate of 92.25%, which means that this application has a good performance in detecting driver drowsiness.

Keywords: Drowsiness, Drowsiness Detection Technology, Facial Landmark, Command Line Interface Application.

1. INTRODUCTION

The number of traffic accidents in Indonesia has increased by more than 80 percent. Where the death toll reached 120 people per day [1]. This data is certainly very concerning because accidents not only cause death but can also cause disability and a large economic burden for victims and their families. Many of the traffic accidents that occur are mostly due to driver fatigue. Drowsiness reduces the level of concentration, activity, and affects mental alertness and decreases the driver's ability to operate the vehicle safely, and increases the risk of human error that can result in fatalities and injuries. Fatigue is defined as drowsiness resulting from neurobiological processes that regulate sleep and circadian rhythms or more simply the urge to sleep [2]. Various measuring instruments that are objective and subjective to assess the level of drowsiness have been developed. However, there are still not many objective measuring instruments that can measure, evaluate and monitor the occurrence of drowsiness in real time, and this is very important to be developed.

The Seybold Report
ISSN 1533-9211
DOI 10.5281/zenodo.7273935

At this time, many drowsiness detection devices have been developed, using diverse devices and applications. Existing detection devices can be categorized into active and passive. Active methods usually use special hardware, which is quite expensive and intrusive, for example, infrared cameras and illuminator [3], [4].

The development of detection devices continues to be carried out by various studies, one of which is an intelligent transportation system based on digital image processing and computer vision. In general, image-based drowsiness detection is to recognize the symptoms of drowsiness on the face, such as yawning mouth, eye blinking, and head nodding. These three parameters have not been used simultaneously in detecting drowsiness, generally using only one or two parameters. A drowsiness detection study by [5], used two parameters, eyelid movement, and facial pose. [6] In his research proposed a system that can detect drowsiness and provide a warning signal when the driver is drowsy to wake up again.

This study aims to develop technology based on intelligent system applications, with digital image processing and computer vision for the driver (user), using three parameters, namely eye blinking, mouth duration when yawning (yawn detection), and head nodding when drowsy (head pose detection), which is designed to detect drowsiness in the driver (user) by receiving input in the form of image capture of the face and providing output in the form of a warning signal (alarm) if the driver is detected drowsiness. Facial area detection uses the facial landmark method which works by allocating points on the face to determine the biological shape of the human face. Processing facial images will use Dlib facial landmarks, OpenCV, and the Python programming language.

In the first stage, drowsiness detection technology based on the command line interface (CLI) application will be developed. The consideration of choosing a CLI-based application stack is based on the effectiveness, speed, and memory efficiency of a technology that adopts the concept of computer vision, where computer vision will require a lot of memory when recognizing images based on the pixels contained in the image. So that developing a program or application, it should be done first based on CLI, so that the selection, determination of programming language, and trials of the operating system can be faster and simpler because it is only in the form of text, which then if it has been successfully developed, it can be used as the basis for developing other programs or applications, for examples such as GUI and Android-based applications.

2. MATERIAL AND METHODS

Facial Landmark

In facial area detection, the facial landmark method is the latest method that works by allocating points on the face to determine the biological shape of the human face [7-10]. Facial landmarks are used to position and represent prominent areas of the face. Facial landmarks consist of eyes, eyebrows, nose, mouth, and jaw lines. The facial landmark can display output in the form of numbers that will function to map parts of the human face as shown in Figure 3. [11].

ISSN 1533-9211
DOI 10.5281/zenodo.7273935

Facial Point Annotation

Used to annotate faces to detect facial landmarks based on the points (coordinates) that have been given to faces [12-14]. This study uses MultiPIE 68 points.

Figure 1: (a)-(d) Annotated images from MultiPIE, XM2VTS, AR, FRGC Ver.2 databases

Helen Dataset

The results of the dataset consisting of 2000 training data and 330 test images with high accuracy, detail, and consistent annotations on the main components of the face [15].

Figure 2: Helen dataset

Dlib Facial Landmark

Dlib is a library that functions by analyzing facial parts with extracting image values, developed by Davis E. King [16]. Extracting the value on the human face dlib will produce a 128-dimensional feature vector [13]. Dlib will be used to help process facial images in the facial landmark. Dlib already has a facial landmark detector [17]. Dlib facial landmarks consist of 68 coordinate labels on the face. Dlib trains the positioning of 68 coordinates using the HELEN data set training data image.

ISSN 1533-9211

DOI 10.5281/zenodo.7273935

Figure 3: Dlib Facial Landmark

Point position with index:

1. Mouth [49, 68]
2. Right Eyebrow [17, 22]
3. Left Eyebrow [22, 27]
4. Right Eye [36, 42]
5. Left Eye [42, 48]
6 Nose [27, 35]
7. Jaw [0, 17]

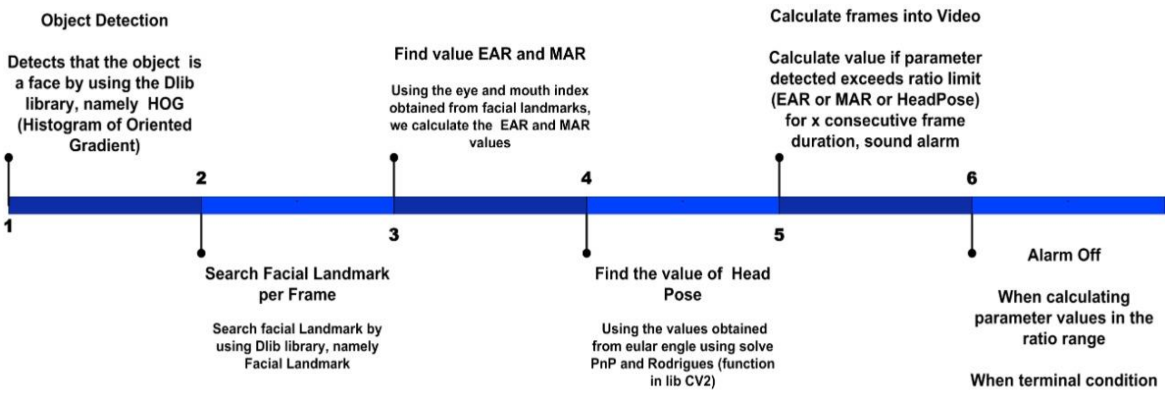
OpenCV (Computer Vision)

OpenCV (Open Source Computer Vision Library) is an open-source computer vision and machine learning software library. OpenCV is a library software open source used for image processing in dynamic real-time. OpenCV can be used in the Python programming language. The algorithm found in OpenCV can be used for face detection, feature matching, and tracking. In OpenCV image processing, it can be used to convert RGB colors to grayscale, display images or videos and other functions instantly [10], [20].

Process Flow

The drowsiness detection process flow is shown in Figure 4.

Figure 4: The drowsiness detection process flow



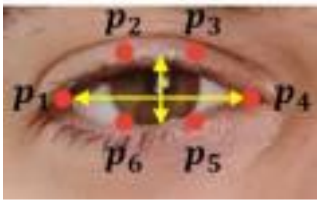
The Seybold Report
ISSN 1533-9211
DOI 10.5281/zenodo.7273935

Detection Algorithm

a. Measurement EAR (Eye Aspect Ratio) [4]

EAR, as the name suggests is the ratio of eye length to eye width. Eye length is calculated by averaging two different vertical lines across the eye as illustrated in the Figure 5.

Figure 5: Eye Aspect Ratio (EAR)



$$\mathrm{EAR} = \frac{\|p_2 - p_6\| + \|p_3 - p_5\|}{2\|p_1 - p_4\|} \quad \ldots\ldots (1)$$

Where p1, …., p6 are 2D locations of landmarks facial. The numerator of the equation is to calculate the distance between the landmarks of the vertical eye while the denominator calculates the distance between the landmarks of the horizontal eyes. When the driver's eyes blink, the aspect ratio of the eyes will decrease drastically, close to zero.

b. Measurement Mouth Aspect Ratio (MAR) [18]

Computationally, MAR is similar to the EAR in that it measures the ratio of the length of the mouth to the width of the mouth. The researchers hypothesize that when drivers are drowsy, they tend to yawn and lose control of their mouths, as in the following Figure 6.

Figure 6: Mouth Aspect Ratio (MAR)

$$\mathrm{MAR} = \frac{|EF|}{|AB|} \quad \ldots\ldots (2)$$

c. Headpose Ratio

Computationally in analyzing a driver's head nod, this study uses the head pose ratio or head vector or head movement when drowsy. The formula used is as follows [19]:

$$R = R_z(\emptyset)R_y(\theta)R_x(\gamma) \quad \ldots\ldots (3)$$

Where (To,θ,γ) = (headpo)

To perform an exact estimation error, it is necessary to calculate MAE (Average Mean Absolute Error) with the following formula.

ISSN 1533-9211
DOI 10.5281/zenodo.7273935

$$\text{MAE} = \frac{1}{N}\sum_{i=1}^{N}\left|\theta_1 - \hat{\theta}_1\right| \quad \text{..... (4)}$$

Accuracy testing

In calculating accuracy using the following calculations:

$$A = \frac{TP + TN}{TP + TN + FP + FN} x\ 100 \quad \text{..... (5)}$$

Where:

TP (True Positive) = The actual condition of drowsiness and a warning that occurs by the system.

FN (False Negative) = The actual condition of drowsiness and does not occur warning by the system.

FP (False Positive) = The actual condition is not sleepy and a warning occurs by the system.

TN (True Negative) = The actual condition is not sleepy and there is no warning by the system.

Dataset and Preprocessing

In this study, the data used was in the form of video images while driving as shown in Figure 5. Videos are taken using smartphone cameras on different drivers and with several lighting conditions, facial conditions while driving, such as blinking, yawning and nodding head when are drowsy.

Figure 5: Video Image Data

3. RESULTS AND DISCUSSION

Drowsiness detection technology based on the developed command line interface (CLI) application can be used on windows and macOS-based PCs and laptops. The application is prioritized to run on Windows with NVIDIA for maximum performance but does not rule out being used on other devices. This command-line interface application can detect drowsiness in real-time using a laptop or computer camera or analyze drowsiness from videos. This application will generate a different warning sound output, based on the detected parameters.

The application_detection.py image display from the video is shown in Figures 6a and 6b.

ISSN 1533-9211
DOI 10.5281/zenodo.7273935

Figure 6a: App_detect.py

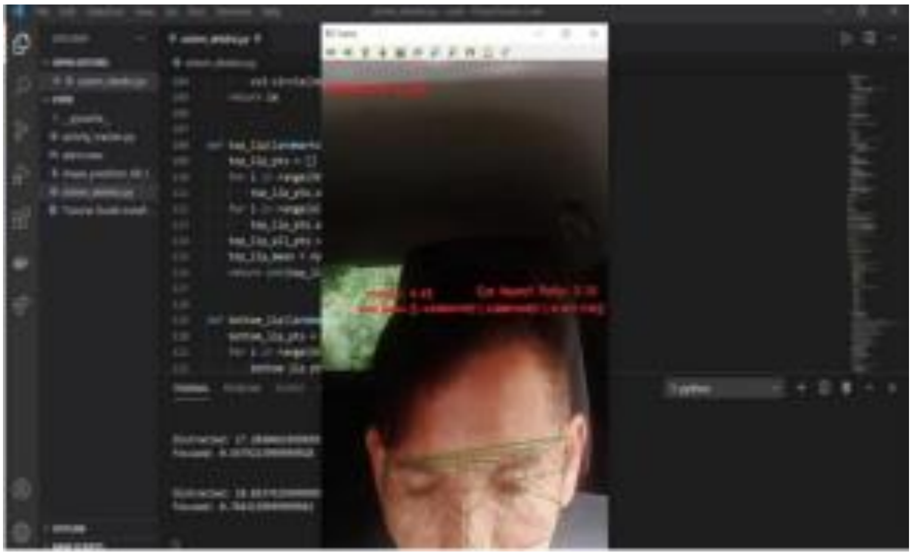
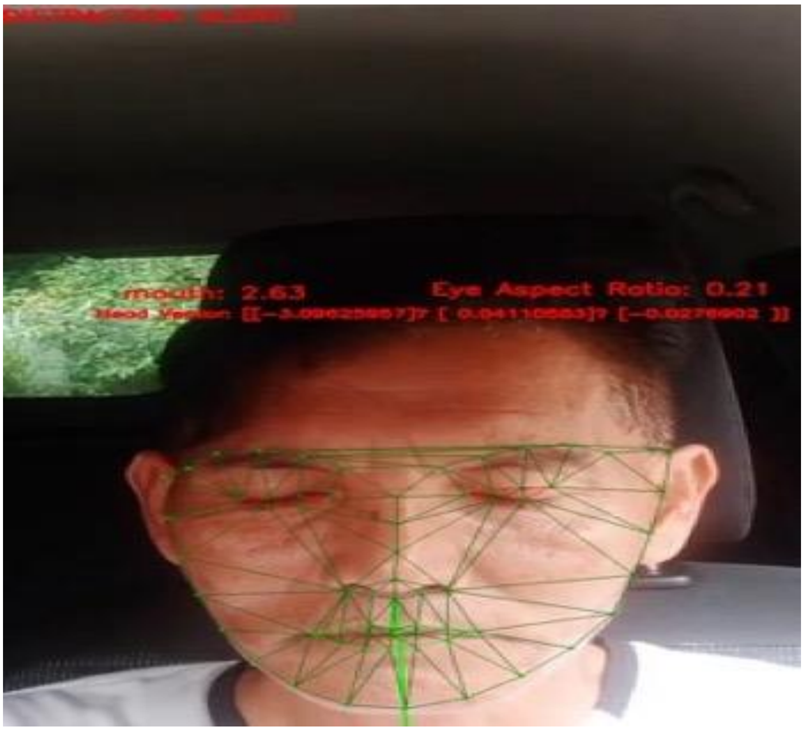


Figure 6b: App_detect.py full image



The accuracy test in this study was tested on 4 different drivers, an example of the detection results can be seen in Figure 7. In the detection of driver drowsiness, if the driver blinks his eyes, yawns, and loses focus or the direction of the head is often nodding, then a warning sound (alarm) will sound to notify the driver to be awake again immediately.

ISSN 1533-9211
DOI 10.5281/zenodo.7273935

Figure 7: Examples of Drowsiness Detection Results

The data on the results of the experiments of each driver, are shown in Table 1 and Table 2. Table 1, the video duration for each driver is different. For results that are filled with the number 0, it means that the system cannot identify anything that is being done, for example in driver 2 with the full video duration 00.05.09, driver 2 is not doing excessive head nodding and because the driver when yawning uses his hand to cover his mouth then it is not detected by the system but driver 2 is drowsiness while driving so it shows the number 19,438.

Table 1: Sample Experimental Data

Respondents	Video Duration	Focused	Drowsy	Yawn
D-1	00.04.50	28.643	22.689	1.129
D-2	00.05.09	0	19.438	0
D-3	00.05.03	5.935	43.038	0
D-4	00.01.40	3.513	5.868	0

Table 2: Accuracy Test Results

Respondents	TP	FN	FP	TN	Accuracy
D-1	10	1	1	0	91, 66 %
D-2	8	5	1	0	92,85 %
D-3	5	6	1	0	91,67 %
D-4	8	5	1	0	92,85 %
Average	92,25 %				

4. CONCLUSIONS

In the first phase of study, drowsiness detection technology based on a command line interface (CLI) application was successfully developed, with an average accuracy test of four drivers resulting in an average accuracy value of 92.25% where these results were declared accurate in detecting driver drowsiness (user). To optimize the drowsiness detection technology, further research is to develop it using other applications, one of which is on a Smartphone, because it will be very useful, in real-time while the driver is driving.

ISSN 1533-9211

DOI 10.5281/zenodo.7273935

Acknowledgements

The authors are grateful to the Directorate General of Higher Education, Research, and Technology-Ministry of Education, Culture, Research, and Technology, Region III Higher Education Service Institute, Mercu Buana University, Udayana University, Bandung Institute of Technology, and all those who helped in this study. This study was also funded by the Domestic Postgraduate Education Scholarship (BPP-DN).

Reference

1. Gita Amanda. (2014). Indonesia ranks first in the increase in traffic accidents. Indonesia Ranks First in Traffic Accident Increase | Republika Online
2. Dawson, D., Searle, AK, & Paterson, JL (2014). Look before you (s) sleep: Evaluating the use of fatigue detection technologies within a fatigue risk management system for the road transport industry. Sleep Medicine Reviews, 18(2), 141–152.
3. Luis M. Bergasa, Jesus Nuevo, Miguel A. Sotelo, Manuel Vazquez. (200$0. Weal Time System for Monitoring Driver Vigilance. IEEE Intelligent Vehicles Symposium University of Parma, 78-83.
4. Tereza Soukupova and Jan Cech. (2016). Real-Time Eye Blink Detection using Facial Landmarks. 21st Computer Vision Winter Workshop: //vision.fe.uni lj.si/cvww2016/proceedings/papers/05.pdf [
5. Ji, Q., & Yang, X. (2002) Real-time eye, gaze, and face pose tracking for monitoring driver vigilance. Real-Time Imaging, 8(5), 357–377.
6. Bharambe, SS and Mahajan, PM (2013). Implementation of Real-Time Driver Drowsiness Detection System. International Journal of Science and Research, 4 (1), 2202-2206.
7. Adrian Rosebrock. (2017a). Eye blink detection with OpenCV, Python, and dlib. Pyimagesearch. https://www.pyimagesearch.com/2017/04/ 24/eye-blink-detection opencv-python-dlib/[
8. Adrian Rosebrock (2017b). Facial landmarks with dlib, OpenCV, Python, and dlib. Pyimagesearch. https://pyimagesearch.com/2017/04/ 03/facial-landmarks-dlib-opencv python/
9. Han, S., Yang, Z., Li, Q., & Chen, Y. (2019). Deformed landmark fitting for sequential faces. Journal of Visual Communication and Image Representation, 381 - 393. [10] Purwanto, AD, & Utaminingrum, F. (2019). Eye Movement and Blink Detection To Select Four Display Menus Using Probability Based Facial Landmarks. 3(11), 10865–10874.
10. Zhong, G. 2019, December 13. Drowsiness Detection with Machine Learning - Towards Data Science. Medium. Towards Data Science. https://towardsdatascience.com/drowsiness-detection-with-machine-learning 765a16ca208a.
11. C. Sagonas, E. Antonakos, G, Tzimiropoulos, S. Zafeiriou, M. Pantic. (2016). 300 Faces In-The-Wild Challenge: database and results. Image and Vision Computing (IMAVIS), 47, 3-18.
12. C. Sagonas, G. Tzimiropoulos, S. Zafeiriou, M. Pantic. (2013). A semi-automatic methodology for facial landmark annotation. Proceedings of IEEE Int'l Conf. Computer Vision and Pattern Recognition (CVPR-W), 5th Workshop on Analysis and Modeling of Faces and Gestures. https://ibug.doc.ic.ac.uk/media/uploads/documents/sagonas_cvpr_2013_amfg_w.pdf
13. C. Sagonas, G. Tzimiropoulos, S. Zafeiriou, M. Pantic. (2013). 300 Faces in-the-Wild Challenge: The first facial landmark localization Challenge. Proceedings of IEEE Int'l Conf. on Computer Vision (ICCV-W), 300 Faces in-the-Wild Challenge (300-W). https://ibug.doc.ic.ac.uk/media/uploads/documents/sagonas_iccv_2013_300_w.pdf
14. Helen dataset. http://www.ifp.illinois.edu/~vuongle2/helen/ . Downloaded February 2020.
15. Niu, G., & Chen, Q. (2018). Learning and video frame-based face detection system for security fields. Journal

ISSN 1533-9211

DOI 10.5281/zenodo.7273935

of Visual Communication and Image Representation, 457 - 463. [17] Kazemi, V and Sullivan, J. (2014). One millisecond face alignment with an ensemble of regression trees. IEEE Conference on Computer Vision and Pattern Recognition, 1867-1874.

16. BK Savaş and Y. Becerikli. (2018). Real Time Driver Fatigue Detection Based on SVM Algorithm. 6th International Conference on Control Engineering & Information Technology (CEIT), 1-4. doi:10.1109/CEIT.2018.8751886.
17. Sabbir Ahmed. (2020). Real-Time Head Pose Estimation with OpenCV and Dlib. https://medium.com/analytics-vidhya/real-time-head-pose-estimation-with-opencv-and dlib-e8dc10d62078.
18. Sugano, H., & Miyamoto, R. (2010). Highly optimized implementation of OpenCV for the Cell Broadband Engine. Computer Vision and Image Understanding, 1273 - 1281.
19. Imultils. https://github.com/PyImageSearch/imutils
20. Budi Sudrajat and Rudianto. (2019). Analysis of the feasibility of information technology investment using the information economics method. Informatics and RPL, 1 (2), 99-105