MATICS: Jurnal Ilmu Komputer dan Teknologi Informasi ISSN : 1978-161X(p); 2477-2550(e)
(*Journal of Computer Science and Information Technology)* *Volume* 15, No. 1(2023), pp 1-7
Received: January 31th 2023; **Accepted**: March 9th 2023 ; **Avalaible Online**: April 15th 2023

Eye Aspect Ratio based on Histogram Oriented Gradient and Linear Support Vector Machine to Microsleep Detection

Mohammad Afinil Maula, Achmad Teguh Wibowo, and Muhammad Andik Izzuddin

Abstract— **Traffic accidents are caused by several factors, especially due to driver fatigue. To minimize accidents caused by human negligence, developing a prototype microsleep detection system to trigger an alarm is necessary. This research uses Histogram Oriented Gradient and Support Vector Machine methods to detect objects. The programming language used is Python, using the dataset from the Idlib face landmark as a marker in the face point area, which will then be calculated based on the eye aspect ratio. This system is implemented using video input captured using a webcam in real-time. The output of this system uses a buzzer to alert the driver. In this study, the test results were obtained well, carried out with 2 test scenarios with a distance of 40cm - 100cm and testing light levels of 33 lux to 226 lux. From these results, the accuracy results were obtained at 88% each.**

Index Terms— **Eye aspect ratio, Microsleep detection, Support vector machine, Histogram Oriented Gradient.**

I. Introduction

NOWADAYS, the development in the computer technology is getting faster; various information and all data can be accessed using a computer. The computers can replace human work because it is more efficient and optimal [1]. The development of technology can have a positive impact on daily human activities. Besides that, there is a negative impact, as the increasing number of accident cases every year in Indonesia from 25.9% to 47.7% [2]. The driver most important role in driving, so it takes extra concentration [3].

Microsleep is the phenomenon of sudden unconsciousness due to drowsiness [4]. During sleepy conditions, the frequency of eye blinks will increase per minute by 20%, so it can be said to be microsleep if the duration of the eyes is closed for 0.5 seconds or higher [5]. So this system is a solution to decrease traffic accidents that often occur due to decreased driver concentration levels. In addition, it can help to monitor drivers to be more vigilant when driving in all conditions [6].

In this research, the development system using python language because a library is a tool that can free to use and access. This research using NodeMCU ESP8266 is needed to design this prototype system, which has a series of chips connecting with a mini computer via a serial port [8].

In previous research, a driver fatigue detection system based on eye blink measurements was created using the viola jones method [9]. Next studies discuss this topic. Among them is research conducted by [10] with the title "Intelligent Driver Drowsiness Detection through Fusion of Yawning and Eye Closure" the research obtained the combination of eye closure and yawning detection methods. The method used for drowsiness detection is fusion-based. Furthermore, research [11] entitled "Driver Drowsiness Detection System Based on Visual Features" in detecting driver fatigue calculated based on eye blinks. The level of drowsiness in general adults is 2-10 seconds per blink of an eye. Then states the indication of a drowsy driver marked by a closed eye blink for 3 or 4 seconds.

From these problems, in this study, the authors will implement a drowsy eye detection system using a different method, HOG + SVM, which is used for face detection. With this method, to get accurate results, this detection system has an estimated time of closed eyes of more than 3 seconds an alarm will sound as a warning.

This paper is arranged as follows: Section II briefly presents the material of Open CV, Histogram Oriented Gradient, Facial Landmark, Support Vector Machine, and many morem, Section III method so in this section. Section IV shows the evaluations. Finally, Section V provides conclusions and directions for future work.

Manuscript received January 31th, 2023. This work was supported in part by Informatics Engineering Department of Maulana Malik Ibrahim Islamic State University.

Mohammad Afinil Maula Author is with Faculty of Science and Technology, UIN Sunan Ampel Surabaya, Indonesia, email H06217011@uinsby.ac.id)

Achmad Teguh Wibowo Author is with Faculty of Science and Technology, UIN Sunan Ampel Surabaya, Indonesia, email atw@uinsby.ac.id)

Muhammad Andik Izzudin Author is with Faculty of Science and Technology, UIN Sunan Ampel Surabaya, Indonesia, email andik@uinsby.ac.id)

II. Materials

A. Machine Learning

Machine Learning is one of the approaches in the field of artificial intelligence that functions to replace or mimic human behavior in solving a problem. Machine learning is intelligently able to mimic the human process of learning [12]. There are two main applications of machine learning, including classification and prediction, where the classification method can sort or classify objects based on specific characteristics. In comparison, the prediction method predicts the data obtained from the test results (training). One of the most popular ML methods is the decision-making system with the Support vector machine and neural network methods [13].

B. Open CV

OpenCV is used for actual-time image processing using open-source licensed libraries. OpenCV is written using the C++ programming language and the availability of interfaces in various programming languages. OpenCV has various applied fields: gesture recognition, face recognition, human and computer interaction, and mobile robotics [14].

C. Support Vector Machine

In 1992, Boser, Guyon, and Vapnik developed a data mining method called Support Vector Machine (SVM). To form a decision line by utilizing the margin, this SVM functions to determine the hyperplane function in the form of a linear model. Margin is the distance between the hyperplane and the nearest data. The SVM method can classify linear models, but it can also solve classification problems in the form of non-linear using concepts such as kernel tricks [15].

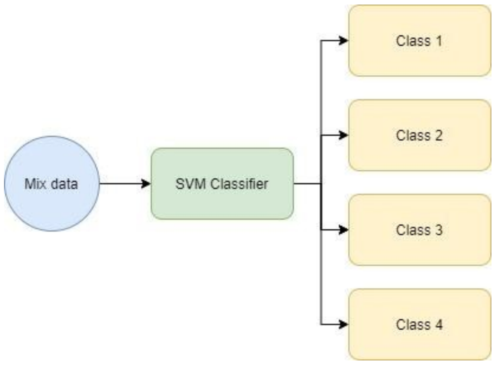

Fig. 1. Support Vector Machine Algorithm

Based on Fig. 1 support vector machine algorithms are widely used for face detection and detect landmark points and landmark points that are detected and then combined into different categorizations, including nose, mouth, eyes, chin eyebrows, cheeks, right-left eyebrows, and lips. The classification process can be used in still images and real-time video capture [16].

D. Histogran Oriented Gradient

Histogram Oriented Gradient (HOG) is a feature extraction used in computer vision and image processing by calculating the gradient value in an image to get the output used to identify. The process of using HOG is that the RGB (Red, Green, Blue) image is converted to grayscale, then the gradient value of each pixel is calculated [17]. Next, determine the number of orientation bins as histogram creation. Then divided into several cells and grouped into larger sizes, commonly called blocks. Furthermore, block normalization is carried out as an R-HOG geometry calculation [18].

E. Facial Landmark

Facial Landmark is an automatic face detector method trained in the dlib library and detects up to 68 (x,y) landmarks as mapping structures on the face [19].

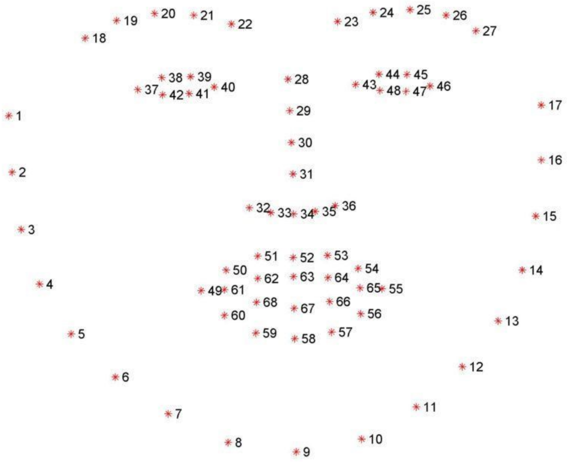

Fig. 2. Facial Landmark Point

Fig. 2 It describes the facial landmark points in 2 different classes, primary and secondary. The first landmarks are the eyebrows, nose, eyes, and mouth. There play an essential role in the facial identity of low-level image features. Secondary landmarks are cheeks, daug, left and proper eyebrows and lips, while the non-extremity point is the nose. In this case, it is crucial to recognize facial expressions and track faces [16].

F. Eye Aspect Ratio

Eye Aspect Ratio (EAR) is predicted to start selecting landmark coordinates. EAR decreases rapidly to zero when sleepy. EAR is the ratio of the vertical to the horizontal length of the eye [20].

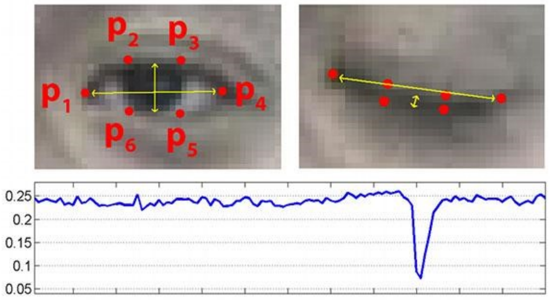

Fig. 3. Eye Aspect Ratio

In Fig. 3, 6 approximate landmarks from the dlib package must be localized to each right or left eye.

These landmarks play a crucial role in calculating the eye aspect ratio according to the guidelines, and there are six landmark points for the eyes, namely P1, P2, P3, P4, P5, and P6, as shown in Figure 6. First, calculate the vertical Euclides distance for the pair of points (P2, P6) & (P3, P5). Next, calculate the horizontal Euclides distance for points (P1, P4) [11].

G. Node MCU

NodeMCU can carry out microcontroller functions and internet connection (Wi-Fi). The tool is an electronic board with the ESP8266 chip base. There are input/output slots, so developing an IOT-based monitoring and controlling application system is easy. The shape of the NodeMCU ESP8266 tool has a USB port that makes it easy to do programming. By using the Arduino IDE, which aims to program through the Arduino compiler [21].

H. Confusion Matrix

The method that can measure the success of a classification method is to compare the classification results performed by the system with the classification results that should be obtained—the confusion matrix method by calculating the percentage of Precision, Recall and Accuracy. There is also a way to calculate the Confusion Matrix using the equation [22].

Table 1. Confusion Matrix Table (Vinet & Zhedanov, 2011)

Predicted Values	Actual Values	
	Positive (1)	Negative (0)
Positive (1)	TP	FP
Negative (0)	FN	TN

Description:
TP (True Positive): the number of correct predictions in the positive class.
FP (False Positive): the number of incorrect predictions in the positive class.
FN (False Negative): number of incorrect predictions in the negative class
TN (True Negative): the number of correct predictions in the negative class.

$$Precision = \frac{TP}{TP+FP} \quad (1)$$

From the above formula, it can be interpreted that Recall (r) is the number of correct predictions in the positive class divided by the number of actual positives in the study.

$$Recall = \frac{TP}{TP+FN} \quad (2)$$

Precision (p) is the number of correct predictions in the positive class divided by the number classified as positive.

$$Precision = \frac{TP+TN}{TP+TN+FP+FN} \quad (3)$$

Accuracy is the ratio of correct predictions (positive and negative) to the total data. So this accuracy can answer the correct predictions from the overall data available [23].

III. Method

The research flow of the microsleep detection prototype is described in the flowchart as follows:

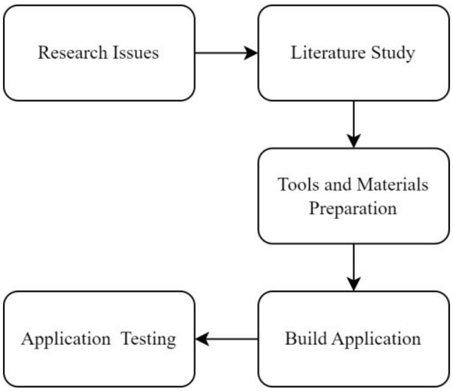
Research Issues
Literature Study
Tools and Materials Preparation
Build Application
Application Testing

Fig. 4. Research Flow

The problem found at this time is the lack of sensitivity to oneself when sleepy or tired, which has an impact that will harm others and oneself, which causes accidents. In this case, an idea emerged to make a prototype work to detect microsleep connected to a microcontroller in the form of NodeMCU ESP8266. In working with this system, the driver's eyes will be captured using a camera in real time. If it takes more than 3 seconds, an alarm will automatically sound.

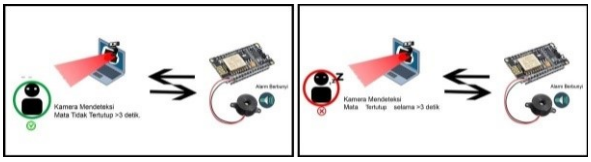

Fig. 5. Microsleep Detection Prototype

For the workings of the microsleep detection prototype, as described in Figure 3.2, the system uses external camera media used to capture the driver's face and requires laptop media used to run the system. Suppose it is detected that the eyes are closed for more than 3 seconds. In that case, the system will send data via the NodeMCU ESP8266 board port serial connection then the buzzer will automatically be triggered so that it sounds an alarm as a warning. Vice versa, if the eyes open again, the alarm will automatically turn off. The stages of making a microsleep detection prototype include:

1. Development System
   To do coding using notepad +++ and run the system using the previous version of the python language, namely 3.8.
2. Importing the ESP8266 library
   When writing code on the NodeMCU ESP8266 board, import the library into the Arduino IDE application. In this case, the Arduino IDE application, by default, cannot connect to the NodeMCU ESP8266.
3. Install serial port 3 COM

Installation is required first on the com3 port to integrate between the laptop and the NodeMCU ESP8266.

4. Writing code on the NodeMCU board

   At this stage, the Arduino IDE application must write the setting code related to the com3 serial port. Then it will be inputted into the NodeMCU board using a micro USB cable connected to a laptop.

After knowing the flow of this research, enter the prototype flow of this microsleep detection system. Related to the flow that will be used will be explained in Fig. 5 below:

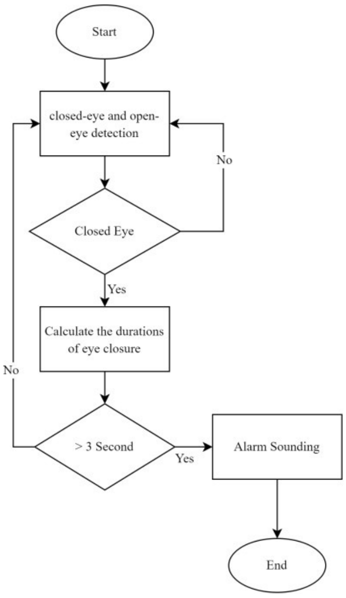
Start
closed-eye and open-eye detection
Closed Eye
No
Yes
Calculate the durations of eye closure
No
> 3 Second
Yes
Alarm Sounding
End

Fig. 5. Flowchart Prototype Microsleep Detection

The workflow of the system includes the following:

1. Start, Running the python program and webcam
2. Using HOG + SVM method to detect eyes
3. Face detection is when the driver's eyes are open or closed using facial landmarks from the dataset from iBug 300-w.
4. If the eyes are closed, the program will calculate the time when the eyes are closed for more than 3 seconds, and the alarm will sound. If less than 3 seconds, the alarm will not sound.

Landmark detection on faces that have 68 specific points on the face is done through several processes. The development process of this microsleep detector includes capturing the face in the image and then detecting landmarks in the face area (eyes). After marking the landmarks in the face area, then calculating the width and height of the eye, calculating the eye aspect ratio (EAR), which is related to the height and width of the eye. Furthermore, it finally displays the number of eye blink duration in the output video as described in Fig. 6.

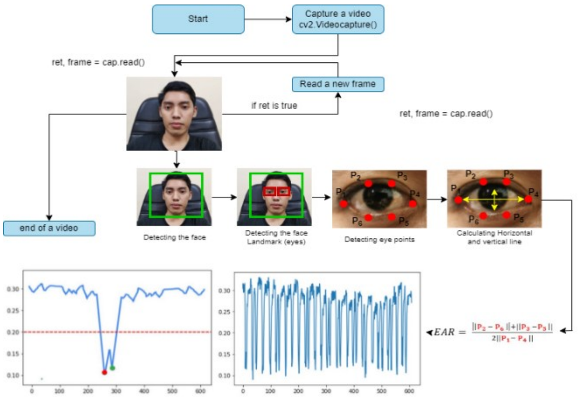

Fig. 6. Eye Aspect Ratio System

In this study using the HOG method for detecting an object (human), there are several stages, including input image using a camera that will be normalized Gamma or convert the colour image to grey.

$$f_0(x,y) = \frac{f_i^R(x,y)+f_i^G(x,y)+f_i^B(x,y)}{3} \quad (4)$$

The calculation changes the colour image into a grey image and then calculates the gradient value of each pixel. Then the number of orientation bins will be determined for histogram creation, called spatial orientation binning. In the gradient computing process, several cells are divided into one larger size called a block. The last process in the HOG algorithm is to normalize the algorithm block, and HOG has detector windows of 64x128 consisting of 8x8 pixels. Then it will be processed by SVM to determine facial features, including eyes, mouth, ears, and nose.

The figure is a block diagram of the microsleep detection flowchart. In this process, capture video with an external camera containing images that will be processed using a python program on the Windows 10 Operating System. Then the output results, whether the driver is sleepy or not, are determined by the eye-aspect ratio.

After going through several stages mentioned above, the next is the creation of microsleep detection systems and applications. Related to this will be explained in the next chapter, in the results and discussion chapter.

IV. Result and Discussion

A. Development System

System programming using the python language and code editor notepad++. The first step is to call the library and function of video capture and face detection and initialize the face predictor. Then start, the video stream capture was taken using a webcam that will be displayed on the system. The detector uses the dlib library, which has been trained based on the modification of the Histogram Oriented Gradient and Support Vector Machine Linear methods to detect objects, then will call the dataset file, which serves to initialize the predictor on the face using the dlib face landmark.

Next is to map the right eye and left eye areas, then calculate the ratio of closed eyes. to get facial landmarks from Idlib, from the data, 68 facial landmarks will then be mapped where the left eye point area is 36, 37, 38, 39, 40, 41, as well as the right eye area, points 42, 43, 44, 45, 46, 47—calling the get_blink_ratio function to calculate the ratio of the two eyes then using the average divided by 2. The time parameter starts when the duration is more than 5.7 seconds. Suppose it is less than 5.7 seconds, then as an end marker. If the parameter is less than -2 seconds, it will trigger an alarm to sound.

B. *Importing the ESP8266 Library*

There are a few steps that need to be done before installing the library, and the first is to set the preferences on file>Preferences>Additional Boards Manager URLs and fill it with "https://arduino.esp8266.com/stable/package_ esp8266com_index.json" as shown in Fig. 7 below:

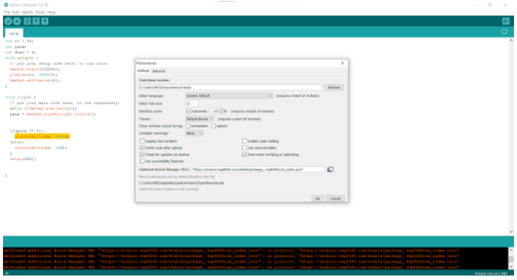

Fig. 7. Preferences Setting

Next, perform tools>board>board manager, as shown in Fig. 8.

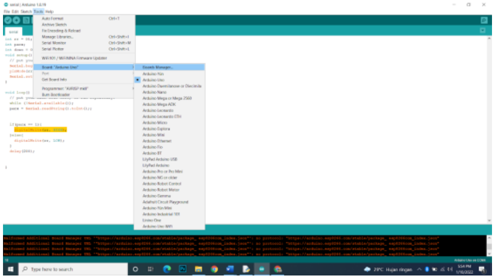

Fig. 8. Manager Board Menu

Furthermore, finally, if you have entered, the board manager will download the library first then the display will appear as in Fig. 9

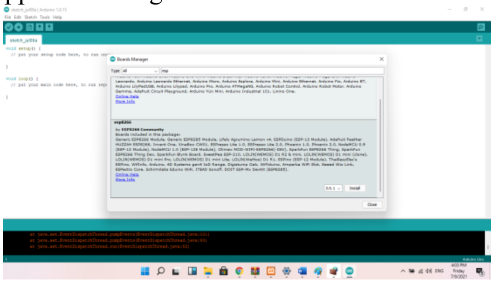

Fig. 9. Boards Manager

Then call the serial to the MCU Node. The serial port line used is COM7 with baudrate = 115200 and timeout = 1, which serves to integrate the buzzer as a warning sound.

C. *Writing code on the NodeMCU board*

The first step is to call the esp8266 library and use port D1 on the Nodemcu ESP8266, and then the second step is writing pseudocode to turn on the alarm, then setting the type of serial used as 115200. Furthermore, a loop is needed to capture if there is a param one call, and the alarm will turn on.

D. *Scenario Testing*

After several stages of designing, the application successfully ran. Related to the display of the results of the microsleep detection system design can be seen in Fig. 10. from the display where the contents of the display detect the face and eyes. In addition, there is a calculation of the results of the distance of the face to the camera. After designing, the next stage is to test the system. for testing itself, distance and light testing will be carried out, and then it will be assessed using the confusion matrix method.

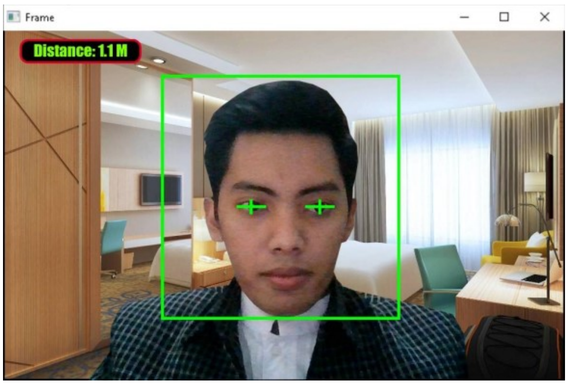

Fig. 9. Microsleep Detection System View

After the manufacture of the microsleep detection system prototype is completed and can be run, then testing is carried out on the system. Testing is done by testing the distance and intensity of light that the system can run. Distance testing is carried out starting from 40 cm to 1 meter. Using a lux meter tool to test light conditions ranging from 33 lux to 226 lux. The light is obtained from the number of lights, four pieces with 5 watts, and 6 room lights that have power up to 144 watts. This test is carried out to determine the success of the microsleep detection system prototype.

Table 2. Scenario Testing Result

Scenario Test	Test Average	Accuracy	Number of Succesfull Tests
Distance Testing	0.074s	89%	168 Test
Intesity of Light Testing	0.05s	89%	120 Test

Based on the results of the test in Table 2 from 2 test scenarios obtained from the results of testing the distance of 40cm - 100cm, the average time obtained is 0.074 s from the number of successful tests of 168 tests getting an accuracy of 89% where from these results it can be evaluated that from that distance the system can detect quickly and accurately, according to the time and accuracy obtained. Furthermore, from testing the light level starting from 33 lux to 226 Lux, the average time obtained is 0.05s from a successful number of 120 tests which get an accuracy of 89%. From the testing results, the light level can be evaluated so that the system can detect light properly and accurately.

Table 3. Confusion Matrix Result

Scenario Test	Recall	Precision	Accuracy
Distance Testing	100%	100%	89%
Intesity of Light Testing	100%	100%	89%

After testing based on the scenario, continue to test the value based on the confusion matrix. What is tested related to Recall (r) is the number of correct positive predictions (True Positive) divided by the actual number of positives in the study. The Precision (p) is the number of correct positive predictions divided by the number of predictions classified as positive. After that, the Accuracy test is the ratio of correct predictions (Positive and negative) divided by the total data. From the above calculations, the value of Recall, Precision is 100%, and Accuracy is 89%, respectively.

V. Conclusion

This research aims to develop a prototype microsleep detection system and measure the accuracy performance based on the Histogram of Oriented Gradient and Support Vector Machine methods in detecting microsleep based on eye aspect ratio. This research is intended to answer the formulation of research problems. The results of this research can be concluded as follows.

1. In developing a microsleep detection prototype to trigger a Linear Histogram Oriented Gradient and Support Vector Machine-based alarm. The tools and materials used are NodeMCU ESP8266, Buzzer, and USB cable. In detecting the stick eye area on the face of the face using the dlib face landmark dataset or from iBug 300 -w. to detect objects using the HOG and SVM methods as classifying objects, after that, it is applied by looking for the eye aspect ratio value which from the calculation results is used to calculate the width and height of the eye as a reference for calculating the duration of closed eyes for more than 3 seconds which will trigger an alarm to sound as a warning to the driver.
2. After developing a prototype microsleep detection system using the Oriented Histogram Gradient + Support Vector Machine Linear method, testing is carried out using a confusion matrix where the test is to determine the performance of the system that has been made. In testing the system using 2 test scenarios, where from the test results obtained, the results of distance testing get an accuracy of 89%, and the test results of the light level testing scenario using a tool in the form of a lux meter obtained an accuracy of 89%. From testing these two scenarios, the performance results of the detection system are accurate.

References

[1] L. R. Cerna, G. Cámara-Chávez, and D. Menotti, “Face detection: Histogram of oriented gradients and bag of feature method,” *Proc. 2013 Int. Conf. Image Process. Comput. Vision, Pattern Recognition, IPCV 2013*, vol. 2, no. July, pp. 657–661, 2013.

[2] P. Chen, “Research on driver fatigue detection strategy based on human eye state,” *Proc. - 2017 Chinese Autom. Congr. CAC 2017*, vol. 2017-January, pp. 619–623, 2017, doi: 10.1109/CAC.2017.8242842.

[3] P. Ii and C. F. Experiment, “Trial begins to investigate efficacy of monoclonal antibody against schizophrenia,” *Pharm. J.*, no. April, 2017, doi: 10.1211/pj.2017.20203911.

[4] C. Aj. Saputra, D. Erwanto, and P. N. Rahayu, “Deteksi Kantuk Pengendara Roda Empat Menggunakan Haar Cascade Classifier Dan Convolutional Neural Network,” *JEECOM J. Electr. Eng. Comput.*, vol. 3, no. 1, pp. 1–7, 2021, doi: 10.33650/jeecom.v3i1.1510.

[5] M. Abdul, R. Irham, and D. A. Prasetya, “Prototipe Pendeteksi Masker Pada Ruangan Wajib Masker Untuk Kendali Pintu Otomatis Berbasis Deep Learning Sebagai Pencegahan Penularan Covid-19,” *Prototipe Pendeteksi Masker Pada Ruangan Wajib Masker Untuk Kendali Pintu Otomatis Berbas. Deep Learn. Sebagai Pencegah. Penularan Covid-19*, pp. 47–55, 2020.

[6] I. Imanuddin, F. Alhadi, R. Oktafian, and A. Ihsan, “Deteksi Mata Mengantuk pada Pengemudi Mobil Menggunakan Metode Viola Jones,” *MATRIK J. Manajemen, Tek. Inform. dan Rekayasa Komput.*, vol. 18, no. 2, pp. 321–329, 2019, doi: 10.30812/matrik.v18i2.389.

[7] M. B. Pranoto, K. N. Ramadhani, and A. Arifianto, “Face Detection System Menggunakan Metode Histogram of Oriented Gradients (HOG) dan Support Vector Machine (SVM) Face Dtection System using Histogram of Oriented Gradients (HOG) Method amd Support Vector Machine (SVM),” *e-Proceeding Eng.*, vol. 4, no. 3, pp. 5038–5045, 2017.

[8] M. R. Hidayat, C. Christiono, and B. S. Sapudin, “PERANCANGAN SISTEM KEAMANAN RUMAH BERBASIS IoT DENGAN NodeMCU ESP8266 MENGGUNAKAN SENSOR PIR HC-SR501 DAN SENSOR SMOKE DETECTOR,” *Kilat*, vol. 7, no. 2, pp. 139–148, 2018, doi: 10.33322/kilat.v7i2.357.

[9] R. A. Putra and F. A. Hermawati, “Sistem Deteksi Kelelahan Pengemudi Berdasarkan Pengukuran Kedipan Mata,” *Konvergensi*, vol. 13, no. 2, 2019, doi: 10.30996/konv.v13i2.2755.

[10] D. Artanto, M. P. Sulistyanto, I. D. Pranowo, and E. E. Pramesta, “Drowsiness detection system based on eye-closure using a low-cost EMG and ESP8266,” *Proc. - 2017 2nd Int. Conf. Inf. Technol. Inf. Syst. Electr. Eng. ICITISEE 2017*, vol. 2018-January, pp. 235–238, 2018, doi: 10.1109/ICITISEE.2017.8285502.

[11] N. N. Pandey and N. B. Muppalaneni, “Real-Time Drowsiness Identification based on Eye State Analysis,” *Proc. - Int. Conf. Artif. Intell. Smart Syst. ICAIS 2021*, pp. 1182–1187, 2021, doi: 10.1109/ICAIS50930.2021.9395975.

[12] S. Ghosh, A. Dasgupta, and A. Swetapadma, “A study on support vector machine based linear and non-linear pattern classification,” *Proc. Int. Conf. Intell. Sustain. Syst. ICISS 2019*, no. Iciss, pp. 24–28, 2019, doi: 10.1109/ISS1.2019.8908018.

[13] A. Ahmad Hania, “Mengenal Artificial Intelligence, Machine Learning, & Deep Learning,” *J. Teknol. Indones.*, vol. 1, no. June, pp. 1–6, 2017.

[14] M. F. Wajdi and J. Sugiantara, “DOI : 10.29408/jit.v1i2.903,” *Infotk J. Inform. dan Teknol.*, vol. 1, no. 2, pp. 96–106, 2018.

[15] P. K. Intan, “Comparison of Kernel Function on Support Vector Machine in Classification of Childbirth,” *J. Mat. “MANTIK,”* vol. 5, no. 2, pp. 90–99, 2019, doi: 10.15642/mantik.2019.5.2.90-99.

[16] A. Jindal and D. R. Priya, “Landmark Points Detection in Case of Human Facial Tracking and Detection,” *Int. J. Eng. Adv. Technol.*, vol. 9, no. 2, pp. 3769–3776, 2019, doi:

10.35940/ijeat.b3367.129219.

[17] R. Y. Endra, A. Cucus, F. N. Afandi, and M. B. Syahputra, “Deteksi Objek Menggunakan Histogram of Oriented Gradient (Hog) Untuk Model Smart Room,” *Explor. J. Sist. Inf. dan Telemat.*, vol. 9, no. 2, 2018, doi: 10.36448/jsit.v9i2.1075.

[18] R. T. Puteri and P. Utaminingrum, “Deteksi Kantuk Menggunakan Kombinasi Haar Cascade dan Convolutional Neural Network,” *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 4, no. 3, pp. 816–821, 2020.

[19] J. P. Tanjung and M. Muhathir, “Classification of facial expressions using SVM and HOG,” *J. Informatics Telecommun. Eng.*, vol. 3, no. 2, pp. 210–215, 2020, doi: 10.31289/jite.v3i2.3182.

[20] A. Kumar, “ISCAIE 2014 - 2014 IEEE Symposium on Computer Applications and Industrial Electronics,” *ISCAIE 2014 - 2014 IEEE Symp. Comput. Appl. Ind. Electron.*, p. 237, 2015.

[21] N. Hidayat, S. Hidayat, N. A. Pramono, and U. Nadirah, “Sistem Deteksi Kebocoran Gas Sederhana Berbasis Arduino Uno,” *Rekayasa*, vol. 13, no. 2, pp. 181–186, 2020, doi: 10.21107/rekayasa.v13i2.6737.

[22] R. Wijanarko and I. Afrianto, “Rancang Bangun Aplikasi Chatbot Media Informasi Parenting Pola Asuh Anak Menggunakan Line,” *Matrix J. Manaj. Teknol. dan Inform.*, vol. 10, no. 1, pp. 1–10, 2020, doi: 10.31940/matrix.v10i1.1805.

[23] G. I. C. G. Ector *et al.*, “The development of a web-based, patient-centered intervention for patients with chronic myeloid leukemia (CMyLife): Design thinking development approach,” *J. Med. Internet Res.*, vol. 22, no. 5, pp. 1–14, 2020, doi: 10.2196/15895.