Drowsiness Detection System for Masked Face Based on Deep Neural Network and Haar Cascade

1st Eryl Nanda Pratama
Department of Computer Science
Telkom University
Bandung, Indonesia
erylnanda@student.telkomuniversity.ac.id

2nd Wikky Fawwaz Al Maki
Department of Computer Science
Telkom University
Bandung, Indonesia
wikkyfawwaz@telkomuniversity.ac.id

2022 1st International Conference on Software Engineering and Information Technology (ICoSEIT) | 978-1-6654-7303-3/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICOSEIT55604.2022.10029948

***Abstract*—Nowadays, technology is growing rapidly followed by modernization. Face detection technology is one technology that has been developed and applied in various sectors such as biometrics recognition systems, retrieval systems, database indexing in digital video, security systems with restricted area access control, video conferencing, and human interaction systems. Eye detection is a further development of face detection in which the image of a human face was detected to be processed by detecting the location of both eyes on the face. Nowadays, the eye detection system can be used as a means of developing more complex applications and can be applied directly in the aspect of technology that uses eye detection like, eye state detection system, drowsiness and fatigue detection system, safety driving support systems or driver assistance system. In this study we propose drowsiness detection system utilizing current novel classification model such as Deep Neural Network (DNN), combined with Haar Cascade. The DNN is utilized to detect face, while Haar Cascade is utilized for detecting the eyes and its state on the detected face. In this study, due to Covid19 pandemic, we focused on developing the classifiers for detecting the face with mask. Apart from that, our proposed classifiers are also capable of identifying non-masked faces. The experimental result showed that by utilizing DNN and Haar Cascade, our proposed system could reach accuracy, precision, recall, and f1 measure as much as 81%, 88%, 80%, and 84%, respectively.**

Keywords— drowsiness detection, Deep Neural Network, Haar Cascade Introduction.

I. INTRODUCTION

Eyes, one of the most salient facial features, have become one of the most important information sources for face analysis. Efficiently and accurately understanding the states of the eyes on a given face image is therefore essential to a wide range of face-related applications, including human computer interface designation, facial expression analysis, driver fatigue detection [1][2], etc.

Previous studies have been conducted in this field. Pauly and Sankar in 2015 developed a drowsiness detection system based on eye blink and tracking [1]. For detecting eye blink and tracking the eyes, Pauly and Sankar utilized Haar-based cascade classifier combined with HOG features and SVM classifiers [1]. The combination of Haar-based cascade classifier combined with HOG features and SVM classifiers could accurately detect the eye blink and tracking the eyes, with the accuracy as much as 94.8%. In 2020, Garg developed a driver's drowsiness detection system using Conventional Computer Vision Application combined with Haar Cascade, EAR and MAR [2]. The result showed the drowsiness detection system with Haar Cascade, EAR and MAR could detect drowsy eyes with 100% accuracy. These results revealed that current technologies could be used for detecting drowsiness on the drivers accurately.

Drowsiness detection is necessary to reduce the number of accidents that occur on the road, especially accidents due to tired and sleepy drivers. However, many drivers are currently donning masks while operating vehicles, particularly in Jakarta, in compliance with regional regulation (PERDA) No. 2 of 2020 [4]. As a result, several studies utilizing yawn detection [2][4] cannot be performed, and practically drowsiness detection in drivers with masked faces has not been the subject of much prior research. Therefore, in this study, we propose Drowsiness Detection System for masked face using DNN combined with Haar Cascade [2][5]. OpenCV DNN has been widely utilized for face detection, which enables real-time detection without consuming a lot of resources. DNN is also capable of detecting both non-masked and masked faces in a variety of poses, while Haar Cascade is utilized for detecting the eyes and its state [6] on the detected face.

II. METHODOLOGY

In this section, we present the details of the proposed face detection system, dataset used, models, and performance metrics. Finally, performance comparison of proposed model compared with another model is shown in the results and discussion section.

A. System Overview

In this study, faces are extracted using DNN, and eyes are extracted using HAAR Cascade and then used to identify whether the eyes are open or closed based on the eyes received from the previous eye detection.

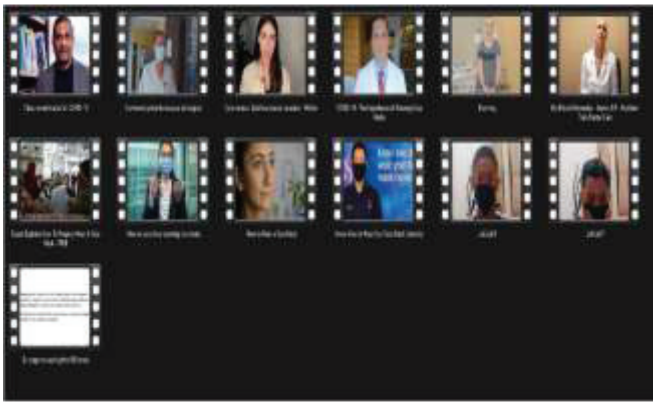
B. Dataset

The dataset used in this study is divided into three different dataset types termed dataset 1, dataset 2, and dataset 3. Dataset 1 consists of a grayscale image dataset without a human face with and without mask or we usually called as background or negative images. This negative image dataset is publicly available dataset which was provided by Muhammad Khalid in “kaggle" dataset [7]:

Fig 1. Negative Images of Dataset 1

For dataset 2, there are 33 videos which are divided into two types: datasets for people who are drowsy and datasets for persons who are not. This dataset was collected personally from random people at the Al-Furqon Islamic Boarding School in Bantarkawung district, Central Java, Indonesia uses a smartphone camera with a resolution of 720 pixels and a duration of about 1 minute, while the others use videos from YouTube. A sample of 12 boys and 10 females were utilized for dataset 2. Figures 2 (a) and (b) display some of the videos used for dataset 2 that were both personally created and obtained from Youtube.

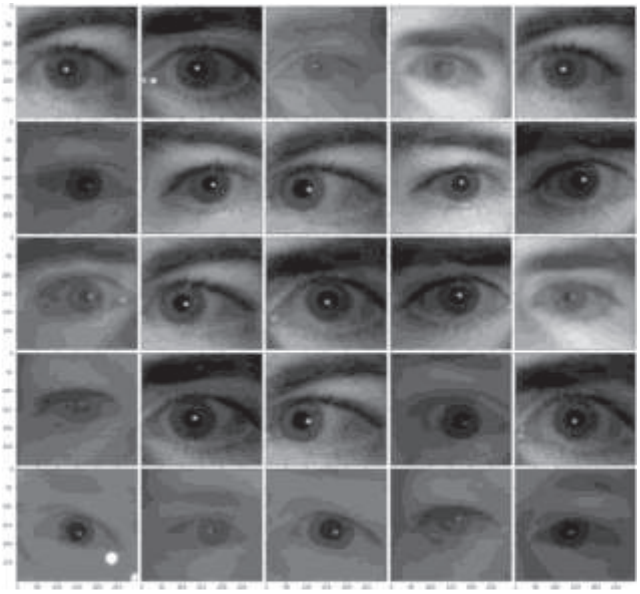
(a)



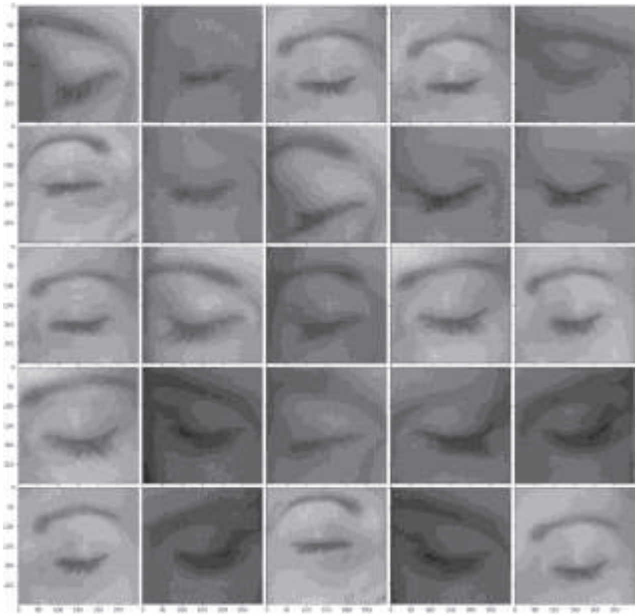
(b)

Fig 2. Video used for Dataset 2 that were Collected Personally (a) Andobtained from Youtube (b)

For dataset 3, a publicly available dataset namely the "Drowsiness Detection Dataset" was utilized. This dataset was provided by Prasad V Patil [8] from “Kaggle” website and consists of eye images that are split into two categories: open and closed eyes. 4000 images in total were utilized in this dataset. The instances of the open and closed eye image in dataset 3 are shown in Figure 3 a and b as follow :



(a)



(b)

Fig 3. The Instances of Open Eye Images (a) and Closed Eye Images (b)in Dataset

C. Deep Neural Network

The Deep Neural Network (DNN) is a component of a distributed representation learning algorithm that looks for the features that make up a multi-level distributed representation in order to extract and organize discriminatory information from data [9][10]. An input layer, a hidden layer, and an output layer, which may be single or multiple are just a few of the fully connected layers that make up a DNN [11]. Information goes from the input layer, through the hidden layer, and toward the output layer in a feed-forward DNN [13]. A feed-forward DNN is illustrated in Figure 4 as follows.

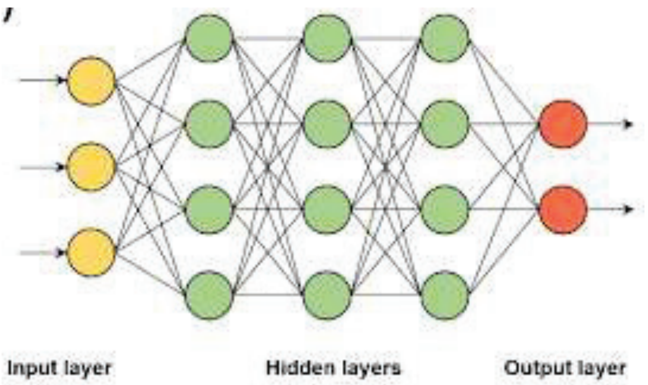


Fig 4. A Feed Forward DNN Illustration

The DNN used in the research is set up in the 'Single Shot Multi-box Detector' (SSD) and uses the 'ResNet-10' architecture as the base model. Single Shot Multi-box Detector similar to YOLO technique which only takes one shot to detect multiple objects in an image using Multibox. It is significantly faster in speed and high-accuracy object detection algorithm. The images from which the model is trained have not been disclosed. Two versions of the model are made available by OpenCV:

1. CaffeImplementation (Floating point 16 version)

2. Original TensorFlow Implementation (8-bit quantized version)

Caffe is a deep learning framework developed as a more faster, powerful and efficient alternative as compared to other object detection methods and is created and managed by Berkeley AI Research (BAIR) and community contributors.We have used the Caffemodel for our implementation to detect faces for drowsiness detection. For this, the Caffemodel and prototxt files were loaded using cv2.dnn.readNet ("path/to/prototxtfile", "path/to/caffemodelweights").

D. Haar Cascade

Haar Cascade is a method for object detection proposed by Paul Viola and Michael James. Haar Cascade is frequently is frequently used to recognize faces and eyes and provides findings that are more accurate than those of other object detection algorithms [5]. However, this method has a weakness which is difficult in identifying faces in images where the faces are neither perpendicular to the camera or facing it [11]. Rectangular black-and-white windows are used by Haar Cascade to extract characteristics [1]. In Haar Cascade, there are four important factors, they are:

1. Haar-like features

Haar-like features are employed to find existing features in an image. The haar-like features method is a square function that gives a specific indication of an image. Haar-like features consist of several rectangular arrangements which are divided into 4 as shown in Figure 5.

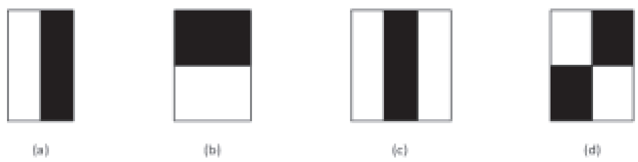


Fig 5. Haar-like Features

As shown in Figure 5, Haar-like features consist of a rectangular arrangement of two rectangles, three rectangles, and four rectangles divided into four features. In Haar-like features, the difference in the number of pixels in the area of the two rectangles is used to calculate the value of the features of two rectangles. The area of the rectangle has the same shape, next to each other vertically or horizontally. For the three-rectangle feature, the calculation is done by adding up the two outer squares, then subtracting the result from the sum of the rectangles in the middle. In rectangular features, the value is obtained from the calculation of the difference between the diagonal pairs [13].

2. Integral Image

Integral image is often used for face detection. After using haar-like features to look for features in the image, Integral image is used to speed up the computational results of taking values from haar-like features. So that the Integral feature is a technique for summing the intensity of pixels in a certain area which can produce a new representation of the previous image. Calculation of the value of one feature can be done quickly by calculating the value of the integral image at 4 points (see in figure 6), the integral image at the location (x,y) contains the number of pixels above and on the left (x,y). With the integral image, the addition of the rectangle of the Haar feature can be carried out by four reference stacks [14].

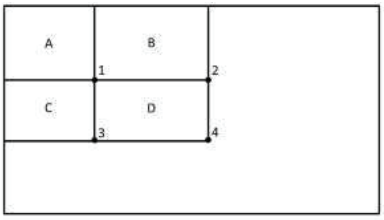


Fig 6. Integral Image Calculation

3. *AdaBoost*

Adaboost is used to enhance classification performance by using small feature sets and training classifiers. The AdaBoost's algorithm initial objective was to enhance poor or simple training procedure's classification performance [14]. Viola Jones suggests utilizing the brute force approach to improve the performance of AdaBoost, which entails first selecting the weakest classifier by scrutinizing each feature in the training data to discover the one that performs the best. AdaBoost produces a powerful classifier when weak classifiers are linearly coupled [14].

4. *Cascade Classifier*

Cascade classifier is used to improve detection performance but reduces computation time [15]. The classification of this algorithm consists of several levels that will produce sub-images that are thought to not be eyes. The structure of Cascade classifier is displayed in figure 7 as follows:

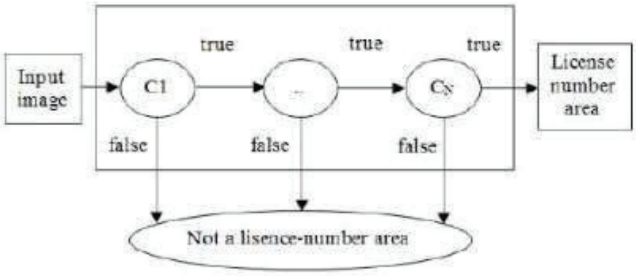


Fig 7. Cascade Classifier Structure

As shown in figure 7, the input image will go to the first level of classification. The results of the strong classifier from Adaboost are used to enter the stratified classification process for each sub-image (image input at each level) with a cascade classifier. Each level's classification will specify whether or not a face is present in the provided sub-image. The sub-image will be discarded if this is the case, preventing it from moving on to the next level [15].

The scope of the study is limited to men or women wearing a face mask and video is taken from head to shoulder using a webcam from laptop Lenovo ThinkPad T460 with 720p resolution in an ideal lightning environment.

In this study, we implement the DNN model, Haar Cascade model proposed in python version 3.7.12 and make use of the sklearn, pandas, cv2, numpy, skimage, os, glob, matplotlib and pylab libraries. We are using Intel(R) Core^{TM} i5-6300U CPU @ 2.40GHz quad core with 8GB RAM. Model evaluation is carried out using three performance evaluation metrics, namely recall, f-score, and accuracy. The formula for calculating performance is shown in Table 1. This study was conducted once with dataset 2 using DNN to extract faces and Haar Cascade to extract eyes and identify whether or not they were open. In this study, the detection of closed eyes for more than or equal to 5 seconds is considered a sign of drowsiness

TABLE I. THE METRICS OF PERFORMANCE EVALUATION

Metrics	Formula
Recall	$\frac{TP}{TP+FN}$
f-score	$\frac{TP}{TP+\frac{1}{2}(FP+FN)}$
Accuracy	$\frac{TP+TN}{TP+FN+FP+TN}$

Notes: *TP*, FP, *FN*, and *TN* are true positive, false positive, false negative and true negative respectively.

III. RESULT AND DISCUSSION

After conducting experiments for drowsiness detection system with DNN to detect faces and Haar Cascade to detect eyes and detect eye conditions, results obtained from the study are shown in Table 1. The result showed that utilized DNN and Haar Cascade as eye detector as well as aye state detector achieved accuracy as much as 81%. The result also shewd that the proposed drowsiness detection system achieved the precision, recall, and f1 score as much as 88%, 80%, and 84%, respectively.

Based on our experiments, DNN is able to detect faces using masks properly in various positions except the position where the face was facing upwards. Haar Cascade is also managed to detect eye conditions well although there were still some detection errors for eyes that were not fully closed. The condition of open and closed eyes can be seen from the box in the face position for green if the eyes are open and red for closed eyes.

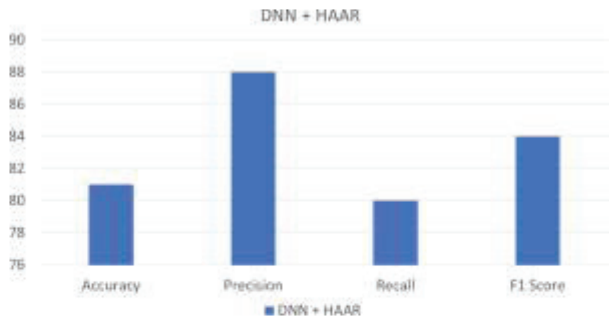


Fig 8. Performance Result of the Proposed Model

Fig 9. Detection Results for Female Video Open Eyes 4 from the Drowsiness Detection System for the DNN and Haar Cascade

Fig 10. Detection Results for Blindfolded Video of Women 4 fromthe Drowsiness Detection System for the DNN and Haar Cascade

IV. CONCLUSION

Based on experimental result of the drowsiness detection system for masked face using DNN and Haar Cascade for detecting face and eyes, as well as its conditions, the system achieved accuracy as much as 81% and f1 score as much as 84% using dataset 2. However, the DNN combined with Haar Cascade methods have lack ability in recognizing a masked face if the position of the face is unstable, such as looking up. This condition is caused of the lower portion of the face is hidden by the mask and the upper part is only dimly visible. Besides of that, the system also has lack ability if eye detection on face area using Haar Cascade fails. The system will detect eye conditions based on its detection from one second prior, thus it makes Haar Cascade un-accurately detect the eye conditions such as detect the open eye as closed eye or vice versa.

REFERENCES

[1] L. Pauly and D. Sankar, “Detection of drowsiness based on HOG features and SVM classifiers,” in 2015 IEEE International Conference on Research in Computational Intelligence and Communication Networks (ICRCICN), Nov. 2015, pp. 181–186. doi: 10.1109/ICRCICN.2015.7434232.

[2] H. Garg, “Drowsiness Detection of a Driver using Conventional Computer Vision Application,” in 2020 International Conference on Power Electronics & IoT Applications in Renewable Energy and its Control (PARC), Feb. 2020, pp. 50–53. doi: 10.1109/PARC49193.2020.236556.

[3] Pemerintah DKI Jakarta, “PERATURAN DAERAH PROVINSI DAERAH KHUSUS. IBUKOTA JAKARTA. NOMOR 2 TAHUN 2020. TENTANG. PENANGGULANGAN. CORONA VIRUS DISEASE 2019,” 2020. https://jdih.jakarta.go.id/uploads/default/produkhukum/PERDA_2_TH_2020.pdf (accessed Mar. 20, 2020).

[4] V. B. Hemadri and U. P. Kulkarni, “Detection of Drowsiness Using Fusion of Yawning and Eyelid Movements,” in Communications in Computer and Information Science, vol. 361 CCIS, 2013, pp. 583–594. doi: 10.1007/978-3-642-36321-4_55.

[5] W. Kongcharoen, S. Nuchitprasitchai, Y. Nilsiam, and J. M. Pearce, “Real-Time Eye State Detection System for Driver Drowsiness Using Convolutional Neural Network,” in 2020 17th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), Jun. 2020, pp. 551–554. doi: 10.1109/ECTI-CON49241.2020.9158265.

[6] N. L. Fitriyani, C.-K. Yang, and M. Syafrudin, “Real-time eye state detection system using haar cascade classifier and circular hough transform,” in 2016 IEEE 5th Global Conference on Consumer Electronics, Oct. 2016, pp. 1–3. doi: 10.1109/GCCE.2016.7800424.

[7] M. KHALID, “Negative Images,” KAGGLE, 2020. https://www.kaggle.com/datasets/muhammadkhalid/negative-images (accessed Jan. 10, 2022).

[8] P. V PATIL, “Drowsiness Detection Dataset,” KAGGLE, 2021. https://www.kaggle.com/datasets/prasadvpatil/mrl-dataset (accessed Jan. 10, 2022).

[9] Y. Bengio, A. Courville, and P. Vincent, “Representation Learning: A Review and New Perspectives,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 35, no. 8, pp. 1798–1828, Aug. 2013, doi: 10.1109/TPAMI.2013.50.

[10] A. Koutsoukas, K. J. Monaghan, X. Li, and J. Huan, “Deep-learning: investigating deep neural networks hyper-parameters and comparison of performance to shallow methods for modeling bioactivity data,” J. Cheminform., vol. 9, no. 1, p. 42, Dec. 2017, doi: 10.1186/s13321-017-0226-y.

[11] I. . Basheer and M. Hajmeer, “Artificial neural networks: fundamentals, computing, design, and application,” J. Microbiol. Methods, vol. 43, no. 1, pp. 3–31, Dec. 2000, doi: 10.1016/S0167-7012(00)00201-3.

[12] T. Danisman, I. M. Bilasco, C. Djeraba, and N. Ihaddadene, “Drowsy driver detection system using eye blink patterns,” in 2010 International Conference on Machine and Web Intelligence, Oct. 2010, pp. 230–233. doi: 10.1109/ICMWI.2010.5648121.

[13] P. Viola and M. Jones, “Rapid object detection using a boosted cascade of simple features,” in Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001, 2001, vol. 1, pp. I-511-I–518. doi: 10.1109/CVPR.2001.990517.

[14] K. Vikram and S. Padmavathi, “Facial parts detection using Viola Jones algorithm,” in 2017 4th International Conference on Advanced Computing and Communication Systems (ICACCS), Jan. 2017, pp. 1–4. doi: 10.1109/ICACCS.2017.8014636.

[15] J. A. Marcial Basilio, G. Aguilar Torres, G. Sanchez Perez, L. K. Toscano Medina, H. M. Perez Meana, and E. E. Hernadez, “Explicit Content Image Detection,” Signal Image Process. An Int. J., vol. 1, no. 2, pp. 47–58, Dec. 2010, doi: 10.5121/sipij.2010.1205.