

Development of Drowsiness Detection System for Drivers using Haar Cascade Classifier and Convolutional Neural Network

Syamsul Mujahidin, Achmad Ripaldi, Bowo Nugroho and Ramadhan Paninggali

Abstract—The use of the Convolutional Neural Network (CNN) method to recognize an object in an image that is not too complex from the background and fore-ground shows very good results. However, in the case of images with various and very complex objects, the CNN method produces a large number of feature maps, sometimes even unnecessary regions of interest (ROI) are included as material for model training which results in a lot of noise. This results in high computational costs and inconsistencies in the prediction results. Therefore, a pre-processing stage is needed, such as determining the area of interest (ROI) on the object of interest and the optimal architecture of CNN. This study applies the Haar Cascade Classifier method to determine the ROI of the object of interest in the image and CNN with the modified vgg-16 model architecture to detect drowsiness in drivers based on facial images. Test results based on the method used show optimal performance on experiments at various epochs with the highest accuracy was achieved 96.72%.

Keywords—Convolutional Neural Networks, Haar Cascade, VGG-16.

I. INTRODUCTION

TRAFFIC accidents can happen anywhere and anytime. Traffic accidents themselves occur due to many factors, such as being distracted while driving, the influence of alcoholic beverages, the condition of the vehicle, and fatigue [1]. Data released by Komite Nasional Keselamatan Transportasi (KNKT) Indonesia states that 80% of traffic accidents that occur in Indonesia are caused by drowsiness and fatigue [2]. Fatigue is often overlooked by drivers, especially when driving long distances. Fatigue makes it difficult for drivers to concentrate while driving. Fatigue also causes microsleep which is very dangerous if it happens to the driver. Microsleep is a condition when a person finds it difficult to keep their eyes open and yawns frequently. Another sign is that he will find it difficult to keep his head upright [1]. Fatigue is often overlooked by drivers, especially when driving long distances. Fatigue makes it difficult for drivers to concentrate while driving. Fatigue also causes microsleep which is very dangerous when it occurs while driving. Microsleep is a condition when a person finds it difficult to keep their eyes open and yawns frequently. Another sign is that it will be difficult for him to keep his head straight.

S. Mujahidin, A. Ripaldi, B. Nugroho, R. Paninggali are with the Department of Informatic, Institut Teknologi Kalimantan, Balikpapan, Indonesia e-mail:syamsul@lecturer.itk.ac.id

Manuscript received July 18, 2023; accepted September 14, 2023.

Research related to drowsiness or microsleep while driving has been carried out by several previous researchers. One of them is Estimation of Driver Vigilance Status Using Real-Time Facial Expression and Deep Learning conducted by Reza etc [3]. This research tries to develop a system for detecting driver's drowsiness by looking at the driver's eyes and mouth. This study utilizes the Haar-Cascade Classifier to extract the eyes and mouth. The architecture used is the LeNet Convolutional Neural Network architecture developed by LeCun researchers and friends. This research itself obtained an accuracy value on the training test of 98%, but for the validation test it obtained an accuracy rate of 84% and it needs development on the input image processing side [3]. The next research was conducted by Shahzeb etc [4] utilizes intrusive technology by looking at the movement of the head and hands using sensors from the XSENS company to capture the movement of these body parts. This research itself utilizes the reLU-BiLSTM architecture to classify driver conditions. This research shows satisfactory results, because the accuracy reaches 99% in certain conditions. This research costs a lot of money plus it can interfere with driver comfort later because it uses a non-intrusive approach, so it cannot be applied commercially [4]. Furthermore, research conducted by Vijay etc [5] utilizes non-intrusive technology, where this research utilizes the Haar-Cascade Classifier in segmenting the eyes and mouth of the driver. Another approach utilizes the Convolutional Neural Network in classifying driver conditions later. This research itself obtained unsatisfactory results, where in the best conditions it obtained an accuracy of 91% and under certain conditions the system could not classify the driver's condition.

The problems in the research previously mentioned have accuracy with other problems such as methods that cannot be realized at this time to problems during real-time testing. This research aims to develop a system for detecting drowsiness or microsleep for drivers with a more optimal model. The Haar-Cascade Classifier method for feature segmentation on the driver's eyes and face, then the classification or learning method utilizes the Convolutional Neural Network with a modified VGG 16 architecture. The program works by detecting the driver's condition at certain time intervals to indicate the driver's drowsiness, if the driver is sleepy, the alarm will sound to remind the driver of the driver's sleepiness. The developed model will be evaluated with the help of a confusion matrix to see the level of accuracy, precision and sensitivity of the model. Evaluation will also use Receiver Operating



Fig. 1: Drowsiness detection datasets

Characteristics and Area Under Curve to see the performance of the model that has been developed. Evaluation is closed by conducting real-time testing to see the system's ability to classify.

II. THE MATERIAL AND METHODS

A. Material

The dataset used is a data set created using MRL and Closed Eyes in Wild (CEW) where the total data used is 2900 images as shown in fig:1. The dataset itself is divided into 4 parts, namely yawning driver, non-yawning driver, closed eyes and open eyes. The open-eye and closed-eye image datasets have various lengths and widths with the RGB color space, and the total images in this dataset are 1484 images. The open face and closed face image dataset has a length and width of 480×640 pixels with RGB color space, and the total images in this dataset are 1452 images. Datasets on the eyes have differences in identity with certain lighting, in contrast to datasets on faces that have diversity such as identity, expression, light contrast in the image, the angle the image is taken, various body postures.

B. Haar-Cascade Classifier

Haar-Cascade Classifier is a framework used to extract features especially on the human face, such as the nose, eyes, mouth and others. The Haar-Cascade Classifier recognizes features that exist on faces through various stages, namely grouping is processed by Haar-Like Features. The goal is to find the desired features whether it's the face itself, nose, eyes and others. Haar-Cascade Classifier works with the help of integral images to speed up the processing of Haar-Like Features. The Haar-Cascade Classifier also utilizes the AdaBoost Algorithm in deciding which features to extract [6].

The Haar-cascade classifier works in a way, first of all it does pooling with the help of the Haar-Like Feature. The Haar-Like Feature will utilize the integral image, so that the Haar-Like Feature can determine the difference between the light

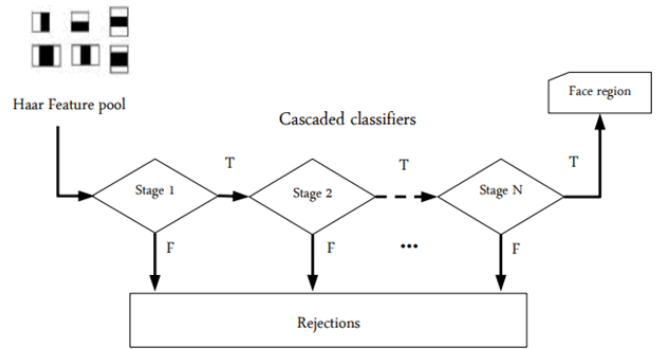


Fig. 2: Haar-Cascade Classifier in determining facial features.

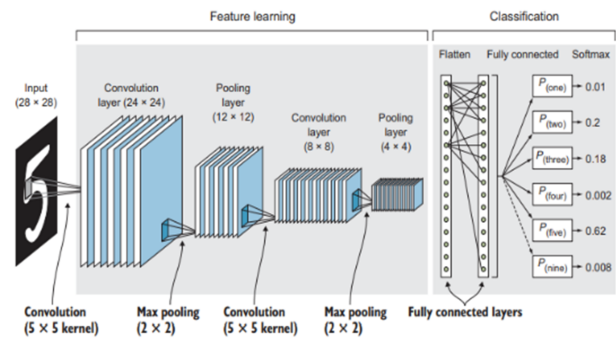


Fig. 3: CNN architecture illustration

intensity area and the dark intensity area value more quickly. The difference in value shown by the Haar-Like Feature will later be used by the Cascade Classifier in determining the feature. Features are extracted in stages according to the rules used in the Cascade Classifier. This stage will involve an additional method, namely the AdaBoost Algorithm. AdaBoost Algorithm aims to compare the results of Haar-Like Features so that the extracted features are appropriate. An overview of the working Haar-Cascade Classifier is shown in fig:2.

C. Convolutional Neural Network (CNN)

Artificial Neural Network is an approach to machine learning that adopts the way the brain's perceptron works to classify a particular object. Artificial Neural Network has a weakness, namely when dealing with data in the form of images, this approach requires handcrafted features to extract the features in the image. The handcrafted features method itself will take quite a long time, because we need to determine what features are relevant to use so that the Artificial Neural Network architecture can classify each new data properly later [7]. These weaknesses eventually emerged a new approach to deep-learning, namely the Convolutional Neural Network, where this approach will extract features and classify simultaneously.

fig:3. shows that there are 2 main processes in CNN namely feature learning and classification. Feature learning is a stage in the Convolutional Neural Network for extracting features in the input image. The result of this stage will be a feature map. Then the Fully Connected Layer is a classification stage on CNN to make predictions about feature maps that have

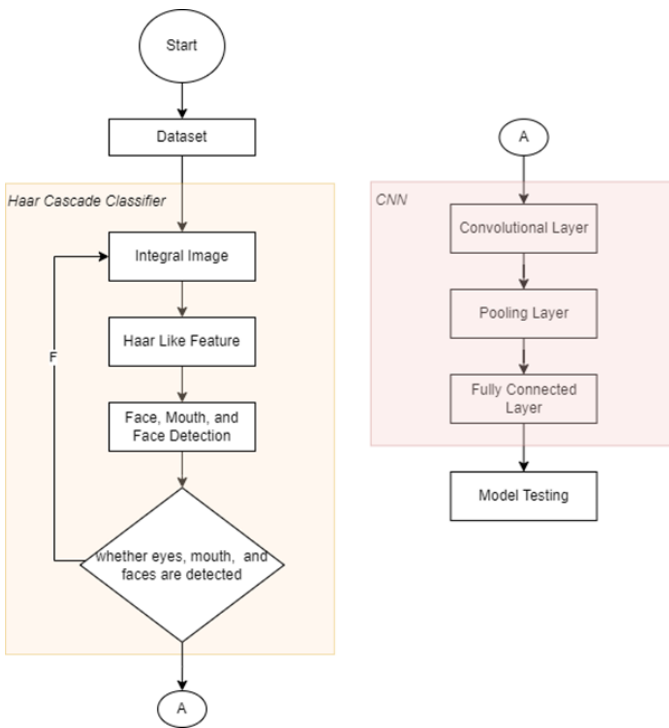


Fig. 4: Flowchart of Drowsiness Detection Method

been processed at the feature learning stage. Classification is a stage for making predictions on feature maps that have been processed at the feature learning stage. This stage requires an additional layer, namely a flatten layer, and this layer will change the previous features into one dimension so that they can be processed on the Fully Connected Layer. The next stage, the features or data will be represented in the form of neurons, and each neuron will have a random weight. Each existing neuron will be studied by the Fully Connected Layer to find certain patterns with the existing classification. The final stage, Fully-Connected Layer will try to predict the class of the image based on the features of the image [8]. To get the best model, there are several parameters that need to be determined before the training process at the fully connected layer stage begins. This parameter is the Learning Rate which functions to determine how far the steps taken by the Fully-Connected Layer are in determining the minimum global point with a certain number of epochs. Weights function to determine the level of role of a neuron or parameter in an output pattern. The optimizer functions to regulate and update the weights of each neuron in this case. The optimizer consists of various methods, one of which is Adaptive Moment Estimation (ADAM), which is a further development of the SGD optimizer which has computational efficiency when dealing with very large amounts of data or parameters [9].

D. Proposed Method

This study utilizes the Haar-Cascade Classifier to classify the eyes, mouth, and faces of drivers. Convolutional Neural Networks are used for feature extraction to classify driver conditions as the system design shown in the fig:4. During the

TABLE I: Comparison of Accuracy Results between Using of Haar-Cascade Classifier and without Haar-Cascade Classifier in CNN VGG 16 Architecture

Epoch	VGG 16	VGG 16 + Haar-Cascade Classifier
35	85.11%	93.11%
50	87.38%	95.38%
100	88.72%	96.72%

segmentation process, the dark side of the image is calculated using the Haar-Like Feature. The results of the Haar-Like Feature are selected with a strong classifier for faces and eyes for BGR images. During the selection process, if the Haar-Like Feature does not match the strong classifier, then the side of the image that is processed by the Haar-Like Feature is not considered part of the ROI on the eye or face. The Convolutional Layer is intended for extracting features assisted by the previous Haar-Cascade Classifier. The architecture developed uses the rules of the Convolutional Layer inspired by VGG Net 16. The number of Convolutional Layers used is 8 layers with a 3 x 3 kernel, and the activation function is ReLu. The number of filters with multiples of 2 is 32, 64, 128 and 256 for each layer. Meanwhile Pooling Layer uses MaxPooling with a kernel size of 2 x 2. The number of layers used is 3 layers and is divided into each existing Convolutional Layer block. The results from the convolutional layer and pooling layer are entered into the flattening stage so that they can be processed by the Fully-Connected Layer. The Fully-Connected Layer section that is designed consists of 3 layers, namely 2 hidden layers and finally the output layer. Each hidden layer has 128 neurons with the activation function used, namely ReLu. The output layer uses 4 neurons that refer to 4 classes from the dataset using the softmax activation function. Visualization of the proposed model is attached in fig:5.

III. RESULTS AND DISCUSSION

The architecture training process was carried out in 35 epochs, 50 epochs, and 100 epochs. The training was conducted with a Ryzen 5 4600H processor and a GTX 1650 Ti GPU. During batch size training, as many as 30 data are used in one iteration. Optimization uses Adam with a learning rate of 0.0001. Augmentation data from training and validation data are used in the model training process. Fig. 6 shows the results of detecting drowsiness in a real time camera using the CNN vgg16 + Haar Cascade Classifier method.

The results of the study show that using the Haar-Cascade Classifier first to extract features from images of eyes, mouths, and faces leads to significantly higher accuracy in CNN training for sleepiness detection than not using the Haar-Cascade Classifier. At epoch 35, the VGG-16 architecture with Haar-Cascade Classifier has an accuracy of 93.11%, while the VGG-16 architecture without Haar-Cascade Classifier has an accuracy of 85.11%. This difference in accuracy is even more pronounced at epoch 100, where the VGG-16 architecture with Haar-Cascade Classifier has an accuracy of 96.72%, while the VGG-16 architecture without Haar-Cascade Classifier has an accuracy of 88.72%.

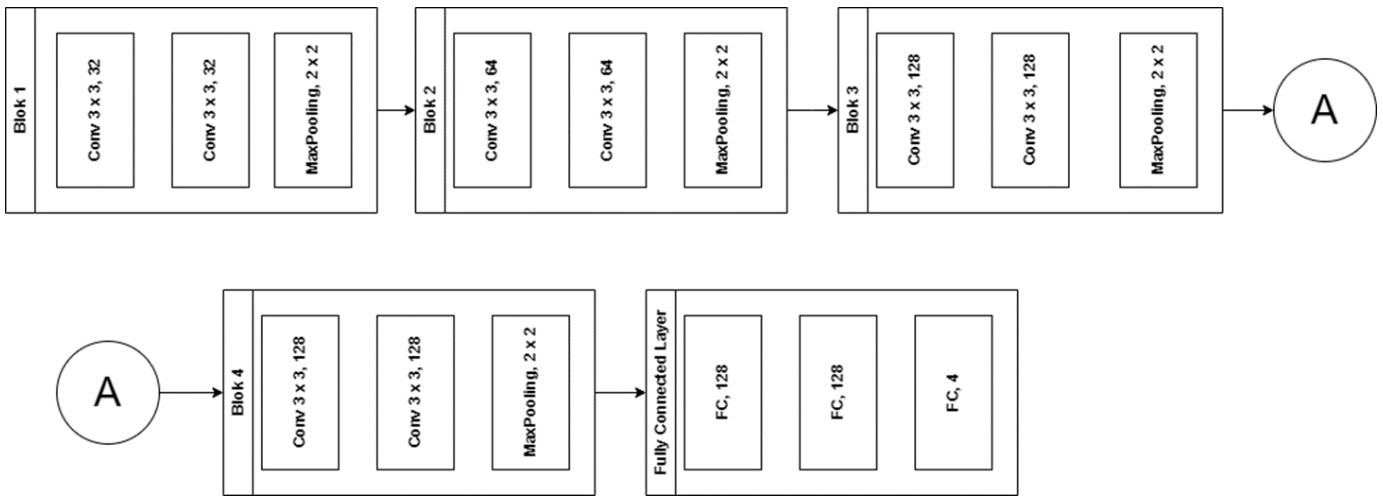


Fig. 5: Modified VGG-16 Architecture

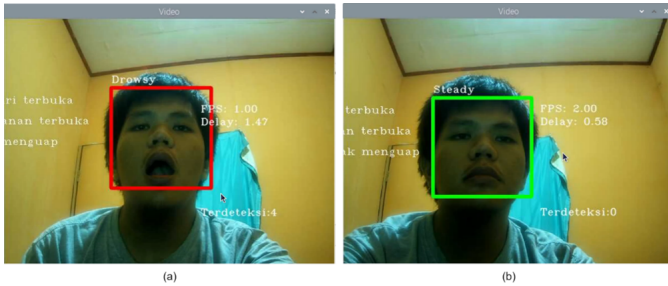


Fig. 6: The results of detecting drowsiness in a real time camera use the CNN vgg16 + Haar Cascade Classifier method

There are a few reasons why using the Haar-Cascade Classifier first can lead to higher accuracy in CNN training for sleepiness detection. First, the Haar-Cascade Classifier is a pre-trained algorithm that can be used to detect specific features in images. In the case of sleepiness detection, the Haar-Cascade Classifier can be used to detect eyes, mouths, and faces. These features are all relevant to sleepiness detection, as they can be used to identify signs of drowsiness, such as closed eyes or drooping eyelids. Second, using the Haar-Cascade Classifier first can save the CNN model the trouble of learning to extract features from scratch. This can be a more difficult task, as the CNN model has to learn to identify the relevant features from the raw image data. This can lead to lower accuracy, as the CNN model may not be able to learn the relevant features as effectively.

Overall, the results of the study show that using the Haar-Cascade Classifier first can be a valuable way to improve the accuracy of CNN training for sleepiness detection. This is because the Haar-Cascade Classifier can extract more relevant features for sleepiness detection, and it can also save the CNN model the trouble of learning to extract features from scratch.

IV. CONCLUSION

This research shows that the author's VGG 16 model or the proposed model is more optimal than the VGG 16 and LeNet models in classifying driver conditions. This conclusion

is from the results of training or evaluation of confusion during testing on two participants. The test results show that the author's VGG 16 model has more optimal performance in terms of accuracy and loss. Evaluation of testing on two participants shows that the 16 writer 50 epoch VGG model has an average accuracy of 96.67%, precision of 94.27% and sensitivity of 93.33% for the first participant.

APPENDIX

Appendixes should appear before the acknowledgment.

ACKNOWLEDGMENT

References are important to the reader; therefore, each citation must be complete and correct. If at all possible, references should be commonly available publications.

REFERENCES

- [1] A. Amodio, M. Ermidoro, D. Maggi, S. Formentin, and S. M. Savaresi, "Automatic detection of driver impairment based on pupillary light reflex," *IEEE transactions on intelligent transportation systems*, vol. 20, no. 8, pp. 3038–3048, 2018.
- [2] I. Imanuddin, F. Alhadi, R. Oktafian, and A. Ihsan, "Deteksi mata mengantuk pada pengemudi mobil menggunakan metode viola jones," *MATRIK: Jurnal Manajemen, Teknik Informatika dan Rekayasa Komputer*, vol. 18, no. 2, pp. 321–329, 2019.
- [3] R. Tamanani, R. Muresan, and A. Al-Dweik, "Estimation of driver vigilance status using real-time facial expression and deep learning," *IEEE Sensors Letters*, vol. 5, no. 5, pp. 1–4, 2021.
- [4] S. Ansari, F. Naghdy, H. Du, and Y. N. Pahnwar, "Driver mental fatigue detection based on head posture using new modified relu-bilstm deep neural network," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 8, pp. 10957–10969, 2021.
- [5] V. P. Sharma, J. S. Yadav, and V. Sharma, "Deep convolutional network based real time fatigue detection and drowsiness alertness system," *Int. J. Electr. Comput. Eng.*, vol. 12, no. 5, pp. 5493–5500, 2022.
- [6] S. Bakheet and A. Al-Hamadi, "A framework for instantaneous driver drowsiness detection based on improved hog features and naïve bayesian classification," *Brain Sciences*, vol. 11, no. 2, p. 240, 2021.
- [7] V. P. Sharma, J. S. Yadav, and V. Sharma, "Deep convolutional network based real time fatigue detection and drowsiness alertness system," *Int. J. Electr. Comput. Eng.*, vol. 12, no. 5, pp. 5493–5500, 2022.
- [8] F. Liu, D. Chen, J. Zhou, and F. Xu, "A review of driver fatigue detection and its advances on the use of rgb-d camera and deep learning," *Engineering Applications of Artificial Intelligence*, vol. 116, p. 105399, 2022.

- [9] A.-C. Phan, T.-N. Trieu, and T.-C. Phan, "Driver drowsiness detection and smart alerting using deep learning and iot," *Internet of Things*, vol. 22, p. 100705, 2023.