Comparing Haar Cascade and YOLOFACE for Region of Interest Classification in Drowsiness Detection

Muhammad Niko Andrean, Guruh Fajar Shidik, Muhammad Naufal*, Farrikh Al Zami, Sri Winarno, Harun Al Azies, Permarna Langgeng Wiecksono Ellwid Putra

Faculty Computer Science, Informatics Engineering Study Program, Dian Nuswantoro University, Semarang, Indonesia
Email: 1nikoandrean2580@gmail.com, 2guruh.fajar@research.dinus.ac.id, 3m.naufal@dsn.dinus.ac.id, 4alzami@dsn.dinus.ac.id, 5sri.winarno@dsn.dinus.ac.id, 6harun.alazies@dsn.dinus.ac.id, 7langgeng86@gmail.com
Correspondence Author Email: m.naufal@dsn.dinus.ac.id

Abstract—Driver drowsiness poses a serious threat to road safety, potentially leading to fatal accidents. Current research often relies on facial features, specific eye components, and the mouth for drowsiness classification. This research aims to compare the accuracy of drowsiness detection in drivers using two different image segmentation methods, namely Haar Cascade and YOLO-face, followed by classification using a decision tree algorithm. The dataset consists of 22,348 images of drowsy driver faces and 19,445 images of non-drowsy driver faces. The segmentation results with YOLO-face prove capable of producing a higher-quality Region of Interest (ROI) and training data in the form of eye images compared to segmentation results using the Haar Cascade method. After undergoing grid search and 10-fold cross-validation processes, the decision tree model achieved the highest accuracy using the entropy parameter, reaching 98.54% for YOLO-face segmentation results and 98.03% for Haar Cascade segmentation results. Despite the slightly higher accuracy of the model utilizing YOLO-face data, the YOLO-face method requires significantly more data processing time compared to the Haar Cascade method. The overall research results indicate that implementing the ROI concept in input images can enhance the focus and accuracy of the system in recognizing signs of drowsiness in drivers.

Keywords: Decision Tree; Drowsiness Detection; Haar Cascade; Region of Interest; Yolo-Face

1. INTRODUCTION

The state of drowsiness, representing a transitional condition between alertness and sleep, characterized by the desire or impulse to sleep, is a major factor that significantly impacts safety, posing the potential for serious injury, even death, and imposing adverse economic consequences[1][2]. Particularly in the context of driving, driving while drowsy has emerged as a leading cause of a substantial number of automobile accidents each year[3]. Accidents resulting from drowsy conditions can account for up to 20% of total accidents deemed serious, illustrating the extent of their impact on road safety threats[4]. Drowsiness often arises due to fatigue levels that far exceed considered safe limits. In this state, individuals may experience a decrease in alertness, difficulty maintaining concentration, and delayed responses to received stimuli[5]. In other words, drowsiness is not merely a matter of wanting to sleep, it can also disrupt an individual's ability to think and act effectively, particularly in situations requiring heightened alertness and rapid responses.

In efforts to reduce accident incidents and enhance road safety, it is crucial to detect signs of drowsiness in drivers. Previous research by Budak et al. attempted to detect drowsiness in drivers using Electroencephalogram (EEG) signals with deep learning LSTM[6]. However, this method faced implementation constraints as it required specialized devices to obtain EEG data, which may not always be available[7]. Therefore, this research will focus more on a computer vision approach to facilitate implementation and identify drowsy conditions in drivers. The computer vision-based approach encompasses two primary methods that can be applied: individual image processing and processing sequences of images or videos[4]. Methods related to processing image or video sequences require greater computational resources and longer processing times. Therefore, within the scope of this research, emphasis will be placed on individual image processing with the aim of reducing computational costs and processing time. Previous studies have explored image processing techniques to detect signs of drowsy driving, such as facial feature analysis of the driver[8][9][10] and detection of the driver's open mouth condition[11]. The facial feature analysis approach can provide an initial indication of whether the driver is drowsy or not. However, this method faces challenges, namely the less distinctive differences between facial features that indicate drowsiness and those that do not. Consequently, utilizing the entire face does not always yield key features that can reliably indicate drowsiness[12]. On the other hand, open mouth detection methods also struggle to distinguish whether the driver's mouth is open due to yawning or speaking. Therefore, in this study, attention will be focused on detecting drowsiness based on the driver’s eye conditions. As outlined in the research conducted by Jahan et al., they employed a deep learning approach using Convolutional Neural Network (CNN) and the Eye Reflection Light (MRL) dataset, containing images of one eye in various conditions, both open and closed, to detect signs of drowsy driving[13]. On the other hand, the research conducted by Farooq and colleagues adopts a similar approach, specifically targeting the detection of drowsy drivers based on the assessment of open or closed eye conditions. This is achieved by employing a Convolutional Neural Network (CNN) algorithm and Haar Cascade technique to identify the eye's location[14]. Although these approaches make significant contributions, detection based on only one aspect of eye conditions may not be reliable enough to ensure whether the driver is
drowsy or not. Moreover, considering only whether the eyes are open or closed may lead to misinterpretations when the driver is blinking.

While existing studies often rely on methods such as entire facial image analysis, open/closed eye assessment, or specialized devices, our research addresses a critical gap. We introduce a vision-based technique concentrating specifically on both eyes of the driver, utilizing Haar cascade and Yolo face for image segmentation. This not only aims to provide a more accurate classification of drowsiness but also showcases a pioneering method to enhance the accuracy and reliability of fatigue detection technology. The comparative analysis of Haar cascade and Yolo face in the segmentation process contributes valuable insights into the effectiveness of each method, informing advancements in drowsiness detection for enhanced road safety.

The utilization of YOLOface and Haar cascade can enhance the accuracy of eye detection in images. The Haar cascade method involves training a Haar-cascade classifier using a dataset of facial images with various backgrounds. The classifier is trained to detect the position of the eyes, which is crucial for accurate face detection in biometric identification systems such as iris recognition[15]. The Haar cascade classifier is known for its reliability and speed in face detection, making it an effective tool for facial detection[16]. By employing a modified Haar cascade algorithm, this approach can detect frontal facial images and obtain the coordinates of the detected face. On the other hand, YOLO (You Only Look Once) is a one-stage object detection algorithm that has gained popularity in face detection classification due to its balance between accuracy and speed. This algorithm is frequently used to enhance face detection classification as it offers real-time detection capabilities and can efficiently handle various face scales[17]. Therefore, to detect the location of both eyes, the Haar Cascade and YOLO-face methods are employed. The adopted classification algorithm is the decision tree. By using a decision tree, it is expected to achieve good accuracy and understand the differences in pre-processing using YOLOface and Haar cascade.

Both face detection methods leverage the concept of Region of Interest (ROI) to confine the search area in images, enabling the evaluation process to focus solely on crucial areas. These techniques have been applied in various papers to detect facial features such as eyes and facial expressions[18]. In Haar cascade, the ROI is obtained by initially detecting the coordinates of the bounding box of the face, after which the system concentrates the search for secondary features, such as eyes, only within that face's ROI. On the other hand, in YOLOFACE, the region of interest (ROI) is identified by detecting the separate locations of the eyes, followed by a padding process to connect them and form a larger box as the ROI encompassing both eyes. Implementing the ROI concept in both methods allows the drowsiness detection system to operate with greater focus, reliability, and accuracy in identifying signs of driver fatigue to prevent accidents due to drowsiness.

The utilization of Decision Trees for drowsiness detection classification is motivated by its ability to provide a simple and interpretable model for decision-making. This method proves effective in selecting relevant features from input data and establishing a hierarchical structure for classifying drowsy eyes. In a study conducted by Ying Yao et al. on Classification of Fatigued and Drunk Driving Based on Decision Tree, promising results were achieved[19]. The advantage of Decision Trees lies in their ability to avoid inherent issues in multivariate regression models and provide a high level of accuracy in classification. Therefore, the use of the decision tree in classifying eye images for drowsiness detection is considered an effective method, and understanding the differences when using YOLOFACE and Haar Cascade in the segmentation process is anticipated.

Our study compares the accuracy of drowsiness detection in drivers using two different image segmentation methods, namely YOLO-face and Haar Cascade, followed by classification using a decision tree. The results indicate that YOLO-face produces a higher quality Region of Interest (ROI) and eye training data, resulting in a decision tree accuracy of 98.54%, whereas the ROI from Haar Cascade yields an accuracy of 98.03%. Despite YOLO-face achieving higher accuracy, this method processes data much slower compared to Haar Cascade. The findings suggest that implementing the ROI concept in input images enhances the focus and accuracy of the system in recognizing signs of drowsiness in drivers.

2. RESEARCH METHODOLOGY

2.1 Research Stages

This section will elucidate the proposed methodology in the research, encompassing several stages to be undertaken, ranging from dataset collection, data pre-processing, machine learning model training, to the evaluation process. In the context of this study, a fusion of Haar Cascade and YOLO-Face techniques will occur within the framework of our proposed machine learning algorithm, specifically the Decision Tree. To identify drowsy drivers, the detailed steps involved in this research will be outlined more comprehensively in Figure 1, which will be presented. In this study, the dataset is divided into two classes, drowsy drivers and non-drowsy drivers. This dataset will undergo a series of preprocessing stages before it can be utilized for model training. The segmentation stage involves multiple steps, including detection using Haar Cascade and YOLO-Face techniques, followed by image cropping based on the detection points to enable a focus on specific areas within the images. In the image preprocessing stage, resizing is performed to ensure uniform dimensions, and pixel intensity normalization is carried out to range from 0 to 1, optimally preparing image data for machine learning model
training. Subsequently, the dataset will be split into two subsets: 80% of the data for model training and 20% for testing. In the context of this research, dataset training will be executed using the Decision Tree algorithm. Grid search and k-fold cross-validation will be employed during training to identify the optimal algorithm parameters for achieving the best results. After the training phase, the model will be evaluated using the testing dataset. This evaluation will include measurements such as accuracy, precision, and recall, and the results will be recorded in the classification report.

![Research Flow Diagram](image)

**Figure 1. Research Flow**

A. Dataset

In this study, the dataset is divided into two distinct classes: drowsy drivers, comprising 22,348 images, and non-drowsy drivers, consisting of 19,445 images, resulting in a total 41,793 images. This dataset has previously been utilized in an effort to detect drowsy drivers through facial feature analysis using Deep Neural Network techniques. Each image in this dataset has uniform dimensions, measuring 227 x 227 pixels, as depicted in Figure 2.

![Dataset Samples](image)

**Figure 2. Dataset Samples**

Figure 2 presents examples of the dataset used in this research, containing images of drivers in drowsy and non-drowsy conditions. As can be observed, the images vary in terms of perspectives, with the drivers' face shown from different angles. Additionally, the dataset includes samples from both female and male individuals. The uniform 227x227 pixel size of images enables consistent processing of visual features during the machine learning model development process in this study.

B. Segmentation

The segmentation process aims to crop the input images only around the eye area to obtain training data consisting of eye images. This study compares two segmentation techniques, namely YOLO-face and Haar Cascade. Both successfully detect the coordinates of the eye locations, but the quality of the resulting image crops differs. YOLO-face segmentation more accurately cuts precisely at the facial frame boundaries, producing eye images with better resolution and detail due to its utilization of Convolutional Neural Network.
Figure 3. Segmentation Haar Cascade Process (a) Original Image (b) Haar Cascade Eyes Detection (c) Cropped Eyes Image

The segmentation involves the detection process using the Haar Cascade method, as illustrated in Figure 3. After successfully identifying the eye locations, the image is cropped based on this detection information, retaining only the portion containing both eyes. Figure 3 illustrates the segmentation process using the Haar Cascade method. It detects the location of the eyes in the original driver image (a). The detected eye positions are depicted by bounding boxes in (b). Based on these coordinates, the image is cropped to retain only the eyes portion (c) for further processing and analysis.

Figure 4. Segmentation Yolo-Face Process (a) Original Image (b) Eye Point (c) Eye Point Padding (d) Cropped Image

Meanwhile, segmentation adopting the YOLO-Face detection approach, as shown in Figure 4, the YOLO-Face segmentation technique individually identifies specific points representing each eye position (b). Padding is applied to these points (c), connecting them to form a bounding region encompassing both eyes. The image is then cropped based on this bounding box to obtain the eye portion (d) for subsequent processing. The main difference lies in the use of YOLO-Face, where eye locations are detected separately at a single point. Padding is applied to this point, and lines are drawn to connect the edges of the padding for each eye, forming a large box encompassing both eye positions. Both segmentation stages will treat the data similarly, using the original data if no eyes are detected in that particular data.

C. Haar Cascade

Haar Cascade is an object detection framework initially introduced through a research paper titled "Rapid Object Detection using a Boosted Cascade of Simple Features" [21]. This algorithm possesses the capability to detect various features in images, such as edges, lines, and corners, utilizing these features to identify objects [22]. Haar Cascade is particularly valuable in face detection as it can recognize unique facial features like eyes, nose, and mouth [23]. The algorithm operates by employing a series of classifiers organized in a cascade structure to detect objects in an image. Each classifier is a machine learning model trained to detect specific features of the target object. These classifiers are arranged sequentially in the cascade, with each classifier becoming more complex and accurate as the cascade depth increases. The algorithm employs a sliding window approach to scan the image, where each window is evaluated by the cascade of classifiers. If the window matches the features of the target object, it is classified as a positive detection. Despite its simplicity, Haar Cascade is capable of real-time execution and does not require intensive computations, making it a common choice in research when detection performance is a determining factor [24].

D. YoloFace

YOLO-Face is a face detection system that leverages deep learning algorithms to identify faces in images and videos [25]. This approach is based on the You Only Look Once (YOLO) algorithm, which is a Convolutional Neural Network (CNN) method capable of performing end-to-end target detection in a single process [26]. The YOLO-Face algorithm incorporates the use of more appropriate anchor boxes for face detection and features a more accurate regression loss function, contributing to its success in efficiently and accurately detecting faces [25]. Several advancements have been made in Yolo-Face, with one of them being Yolo-Facev2, which utilizes the Receptive Field Enhancement module to improve the small receptive field of faces and introduces Repulsion Loss to address face occlusion issues [27].

E. Preprocessing

Data preprocessing is a crucial step in image processing to ensure optimal quality of training data before conducting machine learning model training. There are two main processes carried out in this image data preprocessing stage: resizing and pixel intensity normalization. Resizing aims to ensure uniform pixel...
dimensions across all input images. This is necessary for machine learning algorithms to consistently process features in the images. In the implementation of this study, all images are resized to dimensions of 150 x 50 pixels. Subsequently, pixel intensity normalization is performed on all resized images. This process standardizes pixel intensity values within the range of 0 to 1, facilitating machine learning algorithms in recognizing patterns in the images. Pixel normalization is crucial for improving model accuracy. After the preprocessing process, the dataset is divided into an 80% training subset and a 20% testing subset. With this preprocessing, the image data is optimized for training machine learning models to perform subsequent image classification.

F. Decision Tree

This study proposes the utilization of the Decision Tree algorithm. The classification in this algorithm is constructed by a set of decision nodes and leaf nodes, each indicating decision thresholds and predictions. The operation of the Decision Tree can be observed in Figure 2.5, where, for instance, N and M represent the number of features and predictions, respectively. X = [x1, x2, ..., xN] denotes the feature vector, W = {w1, w2, ..., wN} represents the thresholds, and C = {c1, c2, ..., cM} constitutes a set of predictions[28].

![Figure 5. An Example of Decision Tree Rule Induction](image)

Decision nodes conduct tests on specific features, while leaf nodes represent classification labels[29]. The Decision Tree algorithm separates data based on features that are most effective in distinguishing different classes. The process of forming the Decision Tree begins with the selection of the attribute that will serve as the root of the tree[30]. The primary goal of this attribute selection is to maximize the tree's ability to classify data with unknown classes. The precise choice of the attribute as the root of the decision tree can significantly impact the algorithm's performance and efficiency. If the selected attribute is irrelevant or uninformative, the decision tree may become suboptimal and less effective in classifying new data.

In this study, the Decision Tree algorithm will undergo training using the grid search method to identify the optimal parameters. Grid Search is a technique in the field of Machine Learning that seeks the most optimal combination of parameters for the utilized model. This method explores all potential parameter combinations by dividing the parameter space into a grid. Each parameter combination is assessed based on predefined evaluation metrics. The parameters to be investigated through grid search include the criterion. This criterion determines the method used to measure the quality of separation or feature selection in decision tree formation. The criteria to be tested in both algorithms encompass Gini and entropy. Entropy is employed to measure the impurity level of a dataset, referring to information theory[31]. The entropy calculation can be observed in Equation (1). Conversely, Gini is a parameter used in decision tree algorithms to measure impurity in a set of sample data[32]. Similar to entropy but utilizing the Gini impurity measure, Gini impurity gauges the probability of an error in classifying randomly selected elements in a set if those elements were randomly labeled according to the label distribution in the set[33]. The Gini calculation can be seen in Equation (2).

$$Entropy = \sum_{i=1}^{j} p_i \log_2(p_i)$$  \hspace{1cm} (1)

$$Gini = 1 - \sum_{i=1}^{j} (p_i)^2$$  \hspace{1cm} (2)

Explanation:

$p_i = probability$ of an object being classified into a specific class.

G. Evaluation

In this study, the evaluation encompasses the outcomes of grid search and each fold in the cross-validation process. The objective is to identify the optimal parameters for each algorithm. These optimal parameters will subsequently be applied to the model tested with the testing dataset, and accuracy, precision, and recall metrics will be obtained and recorded in the classification report.

3. RESULT AND DISCUSSION

3.1 Segmentation Process

The segmentation process aims to confine the search area in the input image only to the Region of Interest (ROI), specifically the eye area. This is done to ensure that the system focuses solely on evaluating this crucial area,
enhancing the accuracy and efficiency of drowsiness detection. This study compares the ROI segmentation of eyes using two methods, namely YOLO-face and Haar Cascade. Both techniques also exhibit significant differences in processing time. For the Haar Cascade technique, the entire segmentation process takes 24 minutes and 37 seconds. In contrast, the YOLO-face technique requires a total processing time of 2 hours, 51 minutes, and 04 seconds, significantly longer than Haar Cascade, and the results can be observed in Table 1.

<table>
<thead>
<tr>
<th>Table 1. Segmentation Training Process Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Segmentation process</td>
</tr>
<tr>
<td>Yolo-Face</td>
</tr>
<tr>
<td>Haar Cascade</td>
</tr>
</tbody>
</table>

In Table 1, the significant difference in processing time is attributed to the computational complexity of the respective algorithms employed. Haar Cascade utilizes a straightforward approach based on edge and line pattern recognition to detect facial features. In contrast, YOLO-face employs a deep learning convolutional neural network, which is considerably more intricate and involves intensive tensor computations, leading to a longer image processing time. The total number of successfully segmented data using the Haar Cascade method is 17,312 images for the drowsy class and 14,350 for the non-drowsy class. Meanwhile, with the YOLO-face method, nearly all input images were successfully detected and segmented accurately. The total segmentation results using YOLO-Face are 19,006 images for the drowsy class and 17,306 images for the non-drowsy class. Therefore, both Haar Cascade and YOLO-face fail to achieve complete segmentation for the entire original dataset. This is due to various constraints such as suboptimal face positions, insufficient lighting, and limitations inherent in the segmentation algorithms themselves.

3.2 Decision Tree Training

The training process of the decision tree model in this study was conducted using the Kaggle Notebook platform, a web-based notebook environment specifically designed for machine learning. Kaggle Notebook was chosen because it provides the complete Python programming language along with various popular machine learning libraries and tools such as Pandas, NumPy, SciKit-Learn, PyTorch, and TensorFlow. Additionally, the free access to GPUs provided by Kaggle enables faster model execution. The dataset was loaded into the Kaggle workspace and processed using Kaggle Notebook. The SciKit-Learn library was employed to build and train the decision tree model using the DecisionTreeClassifier function. The GridSearchCV function was used to search for optimal model parameters through 10-fold cross-validation with the accuracy scoring metric. The selection of Kaggle Notebook is expected to expedite model development due to the comprehensive machine learning environment it offers.

The decision tree model training is carried out using 80% of the total dataset as the training data, with the remaining 20% designated for testing. In this study, two dataset categories undergo preprocessing using the Haar Cascade and YOLO-Face methods. The 10-fold cross-validation technique is employed to search for optimal parameter values for the decision tree model. Cross-validation is a standard technique commonly used to evaluate the performance of machine learning models. The method involves dividing the training data into several folds or sections and alternately using these folds for validation and training. Commonly used fold numbers include 2, 5, and 10. In this research, 10-fold cross-validation is chosen as it is deemed to provide a more accurate estimation of model performance compared to fewer folds. With the division into 10 sections, all training data has the opportunity for validation, resulting in smaller variance in accuracy estimation. The average evaluation score across the 10 folds is considered to represent the overall performance of the decision tree algorithm. Therefore, 10-fold cross-validation is deemed suitable for obtaining optimal parameters for the model in this study. During the decision tree training phase, grid search is conducted to find the best parameters for the criteria, accompanied by 10-fold cross-validation for each parameter. In total, 20 model training processes are carried out, and the outcomes of these processes are documented in Table 2.

<table>
<thead>
<tr>
<th>Table 2. Grid Search and Cross Validation Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Preprocess</td>
</tr>
<tr>
<td>-------------</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

Muhammad Niko Andrean, Copyright © 2024, MIB, Page 277
Submitted: 15/12/2023; Accepted: 08/01/2024; Published: 10/01/2024
Tree algorithm may stem from different approaches in image detection and processing. YOLO

The disparity in accuracy between segmentation using the YOLO

3.4

significant. This indicates that the model can effectively detect

differences between the two datasets in terms of accuracy, precision, and recall, these variances are not highly

0.9810. The recall for the drowsy class

lower values. Its accuracy is 0.9803, with a drowsy class precision of 0.9793 and non-drowsy class precision of 0.9810. The recall for the drowsy class is 0.9781, and for the non-drowsy class, it is 0.9821. Despite the numerical differences between the two datasets in terms of accuracy, precision, and recall, these variances are not highly significant. This indicates that the model can effectively detect the drowsiness condition in drivers based on eye conditions, even when using two different segmentation process approaches.

3.4 Analysis

The disparity in accuracy between segmentation using the YOLO-face and Haar Cascade methods in the decision tree algorithm may stem from different approaches in image detection and processing. YOLO-face employs a deep

Table 2 presents the results of grid search and 10-fold cross-validation during the decision tree model training. Grid search was conducted to find the best criterion parameters, namely entropy and Gini, for two types of datasets: the YOLO-face dataset and the Haar Cascade dataset. Subsequently, 10-fold cross-validation was performed to evaluate the model’s accuracy for each of these parameters. The results indicate that the YOLO-face dataset has an average accuracy with the entropy parameter of 0.9876, with an average training time of 10 minutes, and Gini parameter of 0.9837, with an average training time of 16 minutes. On the other hand, the Haar Cascade dataset exhibits a lower average accuracy with the entropy parameter at 0.9815 and an average training time of 9 minutes, and Gini parameter of 0.9797 with an average training time of 24 minutes. This suggests that both YOLO-face and Haar Cascade have the best-performing parameter, which is entropy. These optimal parameters will then be utilized in further testing the model using the testing dataset.

3.3 Testing The Best Parameter

Following the training process utilizing grid search and cross-validation methods, the optimal parameters for each dataset model have been successfully identified. In here the parameters that provided as follows:

1. Max_depth: Maximum depth of the decision tree. The default value of None allows the tree to grow optimally automatically during training, eliminating the need for manual adjustments.
2. Min_samples_split: The minimum number of samples required to split a node. The default value of 2 is generally optimal for most cases.
3. Min_samples_leaf: The minimum number of samples required at a leaf node. The default value of 1 has proven to be effective in many cases.
4. Max_features: The maximum number of features considered for each split. The default value of "auto" adjusts automatically based on the number of features.
5. Max_leaf_nodes: The maximum number of leaf nodes. An unlimited value allows the tree to grow optimally without restrictions during training.
6. Class_weight: Class weights for imbalanced data cases. The default value of none treats all classes equally.

Of all these parameters, ‘criterion’ is the only one that directly and significantly influences the performance of the decision tree as it determines the method for splitting data at each node. Therefore, only the ‘criterion’ is varied in value for the grid search to find the optimal configuration of the model. The next step involves conducting experiments using the best parameters from each dataset on the testing dataset. This testing phase aims to evaluate the performance of models employing the best parameters and obtain more detailed information regarding the accuracy, precision, and recall produced by each model.

Table 2. Grid Search Results

<table>
<thead>
<tr>
<th>Preprocess</th>
<th>Parameter</th>
<th>Fold Score</th>
<th>Fold Score avg (avg.) score training time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dataset</td>
<td></td>
<td>1 2 3 4 5 6 7 8 9 10</td>
<td></td>
</tr>
<tr>
<td>Gini</td>
<td>0.980 0.982 0.975 0.977 0.983 0.977 0.984 0.979 0.983 0.979 0.977 0.979 0.984 0.979 0.979</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Haar Cascade</td>
<td>0.981 0.982 0.975 0.977 0.983 0.977 0.984 0.979 0.983 0.979 0.977 0.979 0.984 0.979 0.979</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

From the information presented in table 3, it is evident that overall, datasets preprocessed with the YOLO-Face method exhibit better performance compared to datasets processed using the Haar Cascade method in the decision tree algorithm model. The YOLO-Face preprocessed dataset shows an accuracy level of 0.9854. In terms of precision, the drowsy class achieves a value of 0.9873, and the non-drowsy class reaches 0.9838. The recall for the drowsy class is 0.9812, while the non-drowsy class attains 0.9890.

Conversely, the dataset that underwent the segmentation process with the Haar Cascade method has slightly lower values. Its accuracy is 0.9803, with a drowsy class precision of 0.9793 and non-drowsy class precision of 0.9810. The recall for the drowsy class is 0.9781, and for the non-drowsy class, it is 0.9821. Despite the numerical differences between the two datasets in terms of accuracy, precision, and recall, these variances are not highly significant. This indicates that the model can effectively detect the drowsiness condition in drivers based on eye conditions, even when using two different segmentation process approaches.

3.4 Analysis

The disparity in accuracy between segmentation using the YOLO-face and Haar Cascade methods in the decision tree algorithm may stem from different approaches in image detection and processing. YOLO-face employs a deep
learning algorithm capable of detecting and recognizing image features at a greater depth than Haar Cascade. In the segmentation stage, this approach may yield higher-quality data, enabling the model to discern the nuances of driver eye conditions more intricately.

![Figure 6. Yolo-Face Preprocessing Result](image)

This is evident in Figure 6, where the segmented data with YOLO-face clearly displays eye condition features, even when the eyes are not aligned (as seen when the head is tilted). Meanwhile, the Haar Cascade method utilizes a simpler face detection approach, focusing on characteristic edge features in images to locate the eyes. The segmentation results using Haar Cascade tend to be less distinct in highlighting the features of drowsy driver eye conditions. The impact of these less detailed segmentation results may be reflected in the model training outcomes, indicating a slightly lower accuracy level.

![Figure 7. Haar Cascade Preprocessing Result](image)

As depicted in Figure 7, data processed using Haar Cascade has smaller sizes, and some data may not clearly capture eye features, especially when the eyes are not aligned. This leads to a lack of clarity in representing specific eye condition features in the segmented data. Nevertheless, Haar Cascade excels in time efficiency during segmentation. The segmentation process of the dataset using the Haar Cascade method takes significantly less time compared to the YOLO-face method. The notable difference in segmentation time can be attributed to the distinct complexities of the two methods. YOLO-face, with its more intricate deep learning algorithm approach, demands more computational resources and time to perform detailed feature detection. This process involves in-depth analysis of each tested frame or image, including more complex processing, thus requiring a longer time. Conversely, Haar Cascade does not necessitate as many computational resources and does not undergo as detailed an analysis process as YOLO-face, resulting in a faster process.

3.5 Discussion

The results achieved in this study demonstrate a high degree of accuracy for drowsiness detection, reaching 98.54% when utilizing the YOLO-face method. This surpasses the outcomes obtained in previous research by Jahan et al. [13], which attained an accuracy of 97.4% for drowsiness classification based on eye conditions using deep learning CNN with a specialized eye image dataset. Similarly, our approach proves superior to the Haar Cascade method for eye detection in drowsy driver identification implemented by Farooq et al. [14], which resulted in an accuracy of 96.7%. The higher performance is likely due to our innovation of considering both eyes instead of a single eye, providing more indicative data features to recognize fatigue.

The segmentation process aims to confine the search area in the input image only around the eyes or the Region of Interest (ROI), allowing the system's evaluation to focus solely on this crucial area. Both YOLO-face and Haar Cascade implement the concept of ROI, albeit with slight differences in approach. Implementing the ROI concept in both methods enables the drowsiness detection system to work more focused and accurately recognize signs of drowsiness in drivers. However, the different workings of the ROI also imply differences in the quality of eye segmentation results. Training data from YOLO-face's ROI has proven to have better resolution and detail compared to Haar Cascade's ROI. It is presumed that this difference in ROI quality contributes to the slightly higher accuracy of the drowsiness detection model based on YOLO-face, at 98.54%, compared to the Haar Cascade-based model with an accuracy of 98.03%. Nevertheless, YOLO-face's superiority in accuracy comes at the cost of significantly longer computation time compared to Haar Cascade. Considering the performance of both methods, the choice of segmentation method should be tailored to the needs of the drowsiness detection system implementation. If highly accurate detection is required, YOLO-face is recommended, while if high computational speed is essential, Haar Cascade is more suitable for implementation.
4. CONCLUSION

The implementation of the Region of Interest (ROI) concept in the image segmentation process, specifically around the eyes, has proven effective in enhancing the focus and accuracy of the system in detecting signs of drowsiness in drivers. The YOLO-face method produces better quality ROI and eye training data compared to Haar Cascade, believed to contribute to the accuracy of the decision tree model for drowsiness detection based on YOLO-face, reaching 0.9854, while the dataset processed with Haar Cascade achieves 0.9803. Both datasets have the same optimal parameter, which is entropy, after undergoing the grid search process. The average accuracy results from cross-validation for both datasets surpass 0.97, indicating good model performance. However, YOLO-face requires a longer time for data segmentation, totaling 2 hours and 51 minutes, while Haar Cascade takes a shorter time, only 24 minutes and 37 seconds.

REFERENCES


