Accredited Ranking SINTA 2
Decree of the Director General of Higher Education, Research, and Technology, No. 158/E/KPT/2021
Validity period from Volume 5 Number 2 of 2021 to Volume 10 Number 1 of 2026

Published online on: **http://jurnal.iaii.or.id**

JURNAL RESTI
(Rekayasa Sistem dan Teknologi Informasi)
Vol. 7 No. 5 (2023) 988 - 995 ISSN Media Electronic: 2580-0760

Fatigue Detection Through Car Driver's Face Using Boosting Local Binary Patterns

Grandhys Setyo Utomo[1], Ema Rachmawati[2*], Febryanti Sthevanie[3]
[1,2,3]Faculty of Informatics, Telkom University, Bandung, Indonesia
[1]emarachmawati@telkomuniversity.ac.id*, [2]grandhysetyo@gmail.com, [3]sthevanie@telkomuniversity.ac.id

Abstract

The general population is concerned with traffic accidents. Driver fatigue is one of the leading causes of car accidents. Several factors, including nighttime driving, sleep deprivation, alcohol consumption, driving on monotonous roads, and drowsy and fatigue-inducing drugs, can contribute to fatigue. This study proposes a facial appearance-based driver fatigue detection system. This is based on the assumption that facial features can be used to identify driver fatigue. We categorize driver conditions into three groups: normal, talking, and yawning. In this study, we used Adaboost to propose Boosting Local Binary Patterns (LBP) to improve the image features of fatigue drivers in the Support Vector Machine (SVM) model. The experimental results indicate that the system's optimal performance achieves an accuracy value of 93.68%, a recall value of 94%, and a precision value of 94%.

Keywords: *fatigue detection; adaboost; boosting local binary patterns (boosting LBP); support vector machine (SVM)*

1. Introduction

At present, the incidence of traffic accidents is a worrying issue among the people of Indonesia. According to Indonesian police statistics, on average, three people die in traffic accidents every hour [1]. The causes of traffic accidents are caused by several factors, namely environmental factors, vehicle factors, and human factors. Human error is the most common cause of traffic accidents. Police data shows that 61% are caused by these factors. This factor is caused by the physical condition of the driver, who experiences decreased focus or fatigue while driving [1]–[6].

Today many physiologists and transport experts have studied the effects of fatigue. It has been established that fatigue is a leading cause of road accidents. This motivates automotive companies to implement in-vehicle intelligent safety systems that assess the driver's state of attention in real-time [7]. This fatigue can be caused by several factors, including nighttime driving, sleep deprivation, alcohol consumption, driving on monotonous roads, and taking drugs that cause fatigue [8].

The driver fatigue detection system must detect whether the driver is tired or sleepy. Measuring driver fatigue is an essential issue because it has several processes [9]. Fatigue detection in previous studies has been developed using techniques for measuring brain waves, heart rate, and pulse [10], [11]. However, these techniques are considered disturbing because the driver must attach the worn device while driving. Several projects that use this technique are carried out by large companies such as Toyota and Nissan; the project is called the Advanced Safety Vehicle (ASV) [12], and the MIT Smart Car project is conducted by the Massachusetts Institute of Technology [13]. Other methods that monitor eye movement and vision using a helmet or special contact lenses have yet to be accepted in clinical settings [14].

One technique that can show fatigue is facial expressions. Facial expression detection can offer more precise detection with minimal effect on the driver [15], [16]. The facial expressions taken are closing the eyes and opening the mouth, which is widely used as a basis for detecting driver fatigue [17], [18]. Facial expressions can be captured using a camera placed in front of the driver. Based on the video image, the system can process the incoming video to conclude the level of driver fatigue [19]. In their yawning detection system, Abtahi et al. [20] utilized the Viola-Jones algorithm for face detection, which represents a better approach than the previous method that relied on alterations in the geometrical characteristics of the mouth [21].

Accepted: 27-12-2022 | Received in revised: 07-08-2023 | Published: 30-08-2023

Pauly and Sankar [22] offer fatigue detection using a Histogram of Oriented Gradient (HOG) feature extraction and Support Vector Machine (SVM) as classification. The paper discusses fatigue detection systems using video images obtained from low-resolution cameras with standard lighting. The system mechanism offered in this paper first inputs the image from the camera, and then the eye detects the image using the Viola-Jones method. Then the image features were extracted using HOG and classified with SVM to determine whether the eye was blinking or open. The results of the system in their paper show that the system with the proposed method has succeeded in detecting human sleepiness. The paper shows that the features in the eye can provide 91.6% accuracy [22]. The researcher concludes that the accuracy value remains high if the lighting is in normal conditions and the video resolution is normal.

Li et al. offer multi-feature fusion and semi-supervised active learning methods [23] in detecting drivers' fatigue. In their paper, the authors combine two features: the driver's facial features and the vehicle's steering features. In their case, the author aims to improve the model's stability and accuracy for detecting driver fatigue. The author conducted several experiments to validate the model. The results show that the accuracy of the model is 86.25% which proves the effectiveness of the driver fatigue detection model. The author concludes their method can run stably when applied and incur low experimental costs.

Yin et al. offer driver fatigue detection based on multiscale dynamic features [24]. At first, the authors used a Gabor filter to get a multiscale representation of the image sequence. Then LBP is used to extract each image. Then LBP is divided into several dynamic units, and the histogram of each dynamic unit is calculated and combined as a dynamic feature. The test results show that the proposed validation approach has an accuracy of 98%. In addition, LBP is very efficient in the feature extraction of an image and has good results in facial expression recognition. LBP can select several parts of the face that contribute to facial expression recognition. LBP also has the most important feature of tolerance to changes in lighting [25], [26].

Recently, Yang et al. [27] utilized a 3D deep learning model for their driver yawning detection system based on recognizing subtle facial movements. They employed 3D convolutional and bidirectional long short-term memory networks to extract spatiotemporal features and used SoftMax to classify data in the YawDD benchmark dataset. Their proposed technique yielded impressive results, achieving an accuracy of 83.4%.

On the other hand, using Multitask Cascaded Convolutional Networks (MTCNN) has been proven to help the system achieve the highest accuracy in the experiments conducted by Deng et al. [31] in the driver drowsiness detection system. They used MTCNN to compensate for the inability of the kernelized correlation filter (KCF) algorithm to mark the face target in the video frame. Their driver-drowsiness detection system extracts the 68 facial features to detect facial regions.

In the meantime, several researchers have demonstrated the advantages of Adaboost [28]–[30]. Wang et al. [28] used Adaboost to learn the base classifiers, which were then used as features and included in the SVM classifier. Xiao et al. [29] also take Adaboost's advantages in dealing with overfitting by combining it with LSTM in a temperature prediction system. Mehmood and Asghar [30] used Adaboost to deal with multiclass classification problems where the features of objects from different classes often overlap. SVM is used as base learners in the proposed ensemble model.

Based on the work of Yang et al. [27], even though the CNN-based approach achieved a pretty good accuracy, it requires a high computational cost in terms of time, resulting in longer training times. It has been demonstrated that hand-crafted feature-based approaches, such as LBP and Adaboost, have relatively short training times and can achieve a competitive level of accuracy. In this paper, we propose the incorporation of MTCNN and Adaboost into a fatigue detection system for in-car drivers based on facial images. MTCNN is applied before the LBP feature extraction procedure to obtain exact face features. Adaboost is used as a boosting technique to improve the model learned from the LBP feature extraction procedure.

This article is structured as follows. In Section 2, we describe the system proposed in this research. In Section 3, we present the results and analysis of the experiment. Finally, we conclude our findings in Section 4.

2. Research Methods

This study's driver fatigue classification system is divided into two stages, namely the training and testing stages, as seen in Figure 1. The video image is entered into the system at the training stage for further preprocessing. Further, LBP is used to extract features from the preprocessing results. The resulting feature vector is then used to develop a classification model using Adaboost with SVM as the base learner. The resulting model is then used in the testing phase to identify driver fatigue.

2.1 Preprocessing

At the pre-processing stage, the video is converted into frames as RGB images. Then the image is converted into a grayscale image. This stage aims to make the data more ideal and as a condition so that images can be processed using the LBP method. Further, face detection is performed using MTCNN Face Detection to obtain the face part on the image [32]. The face detection produces an image of 160 x 160 pixels. At this preprocessing stage, data cleaning removes images that do not match the class label. An illustration of the pre-processing step can be seen in Figure 2.

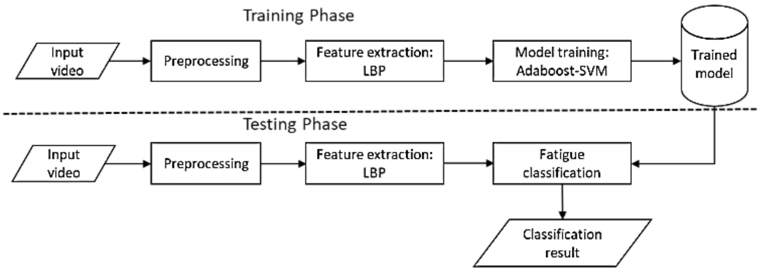


Figure 1. Fatigue detection using Boosting LBP

(a) (b) (c)

Figure 2. Pre-processing: (a) RGB image (b) Grayscale image (c) Face detection result

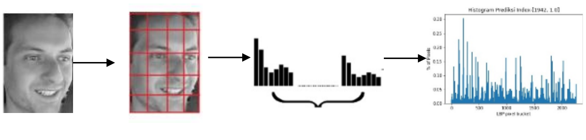
2.2 Feature extraction: LBP

LBP is a non-parametric method that summarizes the image structure by comparing each neighboring pixel. The first thing LBP does is calculate the binary value of each neighboring pixel of the grayscale image. Then the binary values of each neighbor are combined and become the feature values of the LBP. The operator compares each neighboring pixel value as shown in Formula 1.

$$LBP_{P,R}(x_c, y_c) = \sum_{P=0}^{P-1} s(i_p - i_c)2^P \quad (1)$$

$$s(x) = \begin{cases} 1, if\ x \geq 0 \\ 0, if\ x < 0 \end{cases} \quad (2)$$

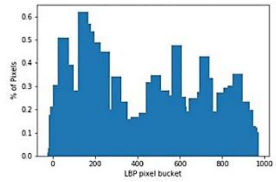
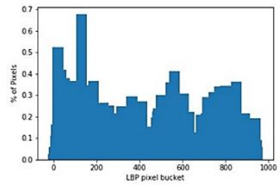
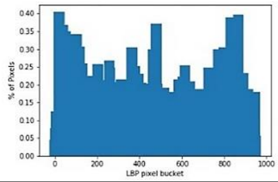
Where P is several sampling points on a circle of radius R, R is a measure of the neighborhood in which a comparative pixel sample is taken for each central pixel in the image, x_c and y_c are the coordinates of the center pixel, i_p is the intensity of the neighboring pixel with index p, i_c is the intensity value of the central pixel c, 2^P is the length of the feature vector. $s(x)$ is a function that converts pixel value into binary [8], as defined in Formula 2. The illustration of the LBP feature extraction process is depicted in Figure 3. Table 1 shows a sample of feature vector histograms in the three classes used in this study.



(a) (b) (c) (d)

Figure 3. LBP feature extraction: (a) face area (b) windows block division (c) histogram of each block (d) feature vector

Table 1. An example of the difference in the histogram of the LBP feature vector in each class

Image	Class	Histogram
	Normal	
	Talking	
	Yawning	

2.3. Adaboost

Boosting is an algorithm used to build an accurate, robust classifier by combining several classifiers because a combination of strong classifiers can achieve very high accuracy [33]. The following is the mechanism that occurs in Adaboost. AdaBoost generates a series of weak classifiers iteratively, with each weak classifier being chosen based on its performance in the training set. Then, in each iterative step, the distribution of weights over the training sample is updated in a manner that forces the weak classifier to prioritize training data that is difficult to classify. This results in a classification with few training errors and excellent generalization performance [34]–[36]. A more detailed process of AdaBoost can be seen in Algorithm 1.

2.4 Support Vector Machine (SVM)

SVM is a data classification method based on statistical learning that can handle regression and pattern

DOI: https://doi.org/10.29207/resti.v7i5.4798

problems. In classifying, SVM produces good accuracy, even better than other machine learning algorithms. The purpose of using the SVM algorithm is to find a hyperplane that can separate two classes with the maximum margin distance.

Adaboost

1. **Given image samples** $(x_1, y_1), \ldots, (x_n, y_n)$ where $y_i = \{0, 1\}$ is negative and positive label, respectively.
2. **Initialize** weight $\omega_{i,j} = \frac{1}{2m}, \frac{1}{2L}$ for $y_i = \{0, 1\}$ where m and l are the number of negative and positive samples, respectively.
3. **Repeat** $t = 1, \ldots, T$:
4. Normalize the weight with $\omega_{i,j} \leftarrow \frac{\omega_{t,i}}{\sum_{j=1}^{n} \omega_{t,i}}$ where ω_i is probability distribution.
5. For each feature j, train classifier where h_j is limited to one feature. Error is evaluated using $\epsilon_j = \sum_i \omega_i \mid h_j(x_i) - y_i \mid$
6. Choose classifier h_t with smallest error ϵ_t
7. Do weight update $\omega_{t+1,i} = \omega_{t,i} \beta_t^{1-e}$ with $\epsilon_i = 0$
8. **if** x_i correctly classified, $\epsilon_i = 1$
9. **else** $\beta_t = \frac{\epsilon_t}{1-\epsilon_t}$
10. Finally, classify $H(x) = \begin{cases} 1, & jika \sum_{T}^{t=1} a_t, h_t(x) \geq 0.5 \\ 0 & \end{cases}$

Algorithm 1. Adaboost

SVM performs implicit data mapping into high feature space and then identifies linear separator hyperplanes with margins to separate data in high dimensional space. The SVM hyperplane maximizes the class imbalance and ensures classification accuracy [25], [34]. A hyperplane can be formed by Formula 3.

$$W.X + b = 0 \quad (3)$$

Where W is the vector weight $\{W_1, W_2, \ldots, W_n\}$, n is the number of attributes, b is the scalar or bias, X is the training tuples.

2.5 Evaluation Metric

Measuring system performance that has been built uses accuracy, precision, and recall values originating from the confusion matrix. The confusion matrix for measuring the performance of the proposed fatigue detection system can be seen in Table 2.

Table 2. Confusion Matrix

Actual label	**Normal**	TN	FNT	FNY
	Talking	FTN	TT	FTY
	Yawning	FYN	FYT	TY
		Normal	**Talking**	**Yawning**
			Predicted Label	

True Normal (TN) is the number of Normal class images predicted as Normal class, True Talking (TT) is the number of images of the Talking class predicted as the Talking class, True Yawning (TY) is the number of Yawning class images predicted as Yawning class, False Normal Talking (FNT) is the number of Normal class images predicted as Talking class, False Normal Yawning (FNY) is the number of Normal class images predicted as Yawning class, False Talking Normal (FTN) is the number of images of the Talking class predicted as the Normal class, False Talking Yawning (FTY) is the number of images of the Talking class predicted as the Yawning class, False Yawning Normal (FYN) is the number of Yawning class images predicted as Normal class, False Yawning Talking (FYT) is the number of Yawning class images predicted as Talking class.

In addition, Euclidian distance was also used in this study to provide information on the distance between feature vectors on feature local discrimination. Euclidean distance is a method used to calculate the similarity of two vectors. The formula for Euclidean distance can be seen in Formula 4.

$$E_{dist} = \sqrt{\sum_{i=1}^{n} (p_i - q_i)^2} \quad (4)$$

Where p dan q are the feature vectors whose distances will be compared.

In this study, Euclidean distance is used to prove prediction errors from images by comparing the average feature vectors of the actual class and then compared with the feature vectors of the predicted class.

3. Results and Discussions

This section describes the description of the dataset used along with the preprocessing performed on the data in Subsection 3.1. In addition, three experimental scenarios are also explained in Subsection 3.2, 3.3, and 3.4, along with the analysis results.

3.1 Dataset and Preprocessing

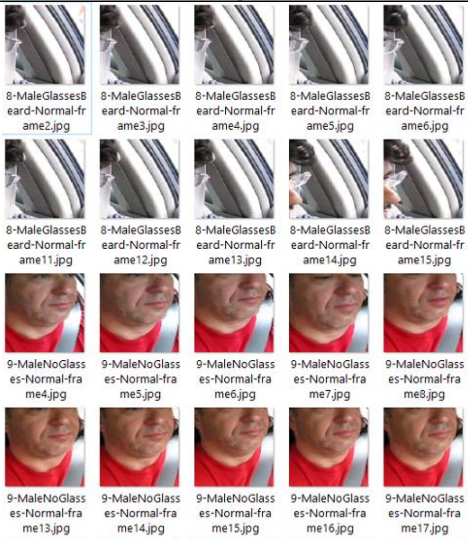
The dataset used in this study is taken from "YawDDataset: A Yawning Detection Dataset" [20], a video image measuring 640x480 pixels with a frame rate of 30 fps. The data comprises 307 videos with facial images facing obliquely to the right. An example of an image from the dataset can be seen in Figure 4.

Figure 4. Sample images from the dataset: (a) Normal (b) Talking (c) Yawning

In the preprocessing step, video data is converted into image frames, producing 5,510 images consisting of 1,937 normal images, 1,875 talking images, and 1,698 yawning images. From the image obtained, several unique characteristics are found in each class. In the normal class, the dominant driver's behavior is silent; there is no change in the mouth. In the talking class, the dominant behavior seen from the driver is a change in the shape of the mouth, and the driver's teeth are visible. Meanwhile, in the yawning class, the driver seemed to yawn, opening his mouth, and his teeth were not visible.

DOI: https://doi.org/10.29207/resti.v7i5.4798

Table 3. Example of images where the face cannot be detected.

Class	Samples
Normal	
Talking	
Yawning	

In this experiment, we only use facial imagery to recognize driver fatigue. Therefore, face detection is performed on the image obtained from the video using MTCNN Face Detection [32]. Several face detection errors were obtained from this process, i.e., the image containing the correct face needed to be obtained. A total of 45 images from the normal class, 14 from the talking class, and 17 from the yawning class did not contain the correct faces. Examples of some incorrectly detected images can be seen in Table 3. Data cleaning was performed manually to detect incorrect face images.

Data cleaning was also carried out on images with inappropriate class labels, as shown in Figure 5. There were 17.9% of the data that was cleaned in this dataset. After preprocessing at this stage, the amount of data is 4,519 images, consisting of 1,877 Normal images, 1,407 Talking images, and 1,311 Yawning images. The experimental data is separated into training and test data by 80% and 20%.

Figure 5. An example of an image from the talking class that does not match the class label.

Meanwhile, the model development in this experiment was carried out in three scenarios to get the best model. Scenario 1 aims to get the best LBP parameters. Scenario 2 seeks to determine the effect of image block size on model performance. Scenario 3 aims to determine the best classifier parameters. The explanation of the experimental results for each scenario is explained in the next section.

3.2 Scenario 1: LBP Feature Parameter Tuning

In the first scenario, the thing to do is to select the LBP feature parameters, namely the P and R parameters (see Formula 1). K-fold cross-validation with **k=10** and SVM was applied to the training data. K-fold cross-validation divides the training data into k parts, with k-1 parts used as training data and one part validation data. The results of the best validation accuracy values for each LBP parameter can be seen in Table 4.

Table 4. The effect of LBP parameters on the model performance. Accuracy (Acc), Precision (Pr), and Recall (Re) values are shown in %. L=vector length. Time=running time in minutes

(P, R)	Acc	Pr	Re	L	Time (m)
(4,1)	**44.33**	36.10	44.33	**16**	0.23
(8,1)	43.38	39.16	43.38	256	2.24
(4,2)	43.14	35.58	43.14	16	0.23
(8,2)	43.66	58.92	43.66	256	2.24
(12,2)	41.53	42.12	41.53	4096	33.35

Table 4 shows that the best validation accuracy value for each feature extraction parameter is LBP with a value of **P=4**. Based on the performance results of the model built in Table 4, it can be seen that the best accuracy is obtained with parameters **P=4 and R=1** with a **vector length of 16**. This happens because, with

DOI: https://doi.org/10.29207/resti.v7i5.4798

these parameters, texture information can already provide good information. Furthermore, feature selection using Adaboost is applied to this combination of parameters, with the results shown in Table 5.

Table 5. The effect of boosting on the model performance. Accuracy (Acc), Precision (Pr), and Recall (Re) values are shown in %. L= vector length. Ls= vector length after boosting. Time=running time in minutes

(P, R)	Acc	Pr	Re	L	Ls	Time (m)
(4,1)	73.09	73.37	73.09	400	44	0.8
(8,1)	70.62	71.02	70.62	6400	49	1.4
(4,2)	**75.84**	**76.49**	**75.84**	**400**	**49**	**0.75**
(8,2)	68.85	70.28	68.85	6400	49	1.38
(12,2)	68.28	69.04	68.28	102400	49	12.7

It can be seen in Table 5 that the parameters for the model with the best performance are obtained at **P=4 and R=2**. This differs from the performance results in Table 4, where the best model performance is obtained using **P=4 and R=1**. In Table 5, it can also be seen that the length of the feature vector as a result of feature selection produced on the use of **P=4 and R=1** contains less feature information when compared to the use of **P=4** and **R=2** so that the performance of using LBP with **P=4** and **R=2** can give better results.

Table 6. The effect of image block size on model performance. Accuracy (Acc), Precision (Pr), and Recall (Re) values are shown in %. L=vector length. Ls= vector length after boosting. Time=running time in minutes

Block size	Acc	Pr	Re	L	Ls	Time (m)
160x160	43.19	42.81	43.19	16	-	0.39
80x80	59.27	60.25	59.27	64	36	0.40
40x40	73.12	73.37	73.12	256	42	0.60
32x32	75.84	76.49	75.84	400	49	0.75
20x20	**79.51**	**79.69**	**79.51**	**1024**	**48**	**1.39**
10x10	74.48	74.69	74.48	4096	50	4.8

3.3 Scenario 2: Experiment with Image Block Size

After selecting the best LBP feature parameters in scenario 1, the image block size is chosen to determine the effect on model performance. The image block size parameters tested were **160x160, 80x80, 40x40, 32x32, 20x20, and 10x10**. Using SVM as a classifier, the model performance results of various image block sizes can be seen in Table 6.

It can be seen from Table 6 that the best accuracy value of the model is obtained at an image block size of **20x20** with a vector length of **1024**. The time required to run the program at this block size is **1.39 minutes**. Applying LBP boosting to the image block size can improve model accuracy better than other image block sizes. This happens because the appropriate image block size and the number of image blocks produced, the better the information generated from LBP feature extraction.

3.4 Scenario 3: Experiment on SVM classifiers

After obtaining the best parameter values and image block sizes from the LBP Boosting feature extraction, hyperparameter settings for Adaboost are performed. The parameters obtained were **n_estimator** of **10000** and **learning_rate** of **1**. Meanwhile, three kernels were tested using the SVM classifier, with the best results obtained on polynomial kernels. The results of the accuracy comparison using the SVM kernel can be seen in Table 7. The best parameters of the SVM are **gamma=1** and **C=0.1**.

Table 7. The effect of SVM kernel(s) on model performance

	Linear	Polynomial	RBF
Acc	81.33%	**89.63%**	89.47%

The performance results of the best AdaBoost and SVM hyperparameters obtained can be seen in Table 8. Based on the model performance results in Table 8, the AdaBoost hyperparameter can increase the model's accuracy performance by 10% from the initial accuracy of 79.51% (see Table 6) to 89.63%. Likewise, the precision and recall values have increased by 10% (see Table 6).

Table 8. Model performance after Adaboost and SVM hyperparameter tuning. LBP parameters (P=4, R=2); image block size=20x20; SVM kernel= polynomial. Accuracy (Acc), Precision (Pr), and Recall (Re) values are shown in %. L=vector length.

Parameter	Acc	Pr	Re	L
LBP (4, 2) Blok 20x20 + SVM Polynomial	89.63	89.75	89.63	1024

3.5 Testing

The test was carried out on 919 test data images. After getting the model with the best performance from the training data, testing is done. Figure 6 shows the confusion matrix from the test results. The test results obtained an accuracy performance of **93.68%**, with a precision and recall value of 94%, as shown in Table 9.

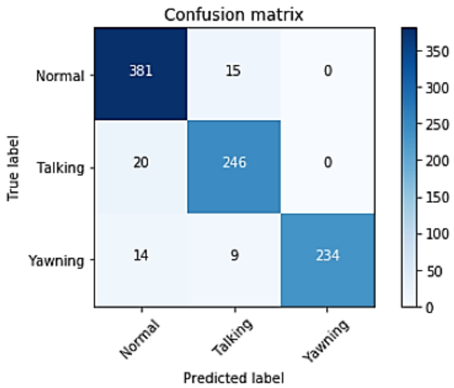


Figure 6. Confusion Matrix

Table 9. System Performance Evaluation

Class	Pr	Re	#images
Normal	0.92	0.96	396
Talking	0.91	0.92	266
Yawning	1.0	0.91	257
Weighted Avg	**0.94**	**0.94**	919

In Figure 6, 15 Normal class images are predicted as Talking, 20 images of the Talking class are predicted as

DOI: https://doi.org/10.29207/resti.v7i5.4798

Normal, 14 images of the Yawning class are predicted as Normal, and nine images of the Yawning class are predicted as Talking. Meanwhile, there is no mispredicted image in the Yawning class. This shows that the system can detect drivers with Yawning conditions.

Figure 7 and Figure 8 show an example of an incorrectly predicted image. The Yawning class image in Figure 7 is incorrectly predicted as a Normal class. Meanwhile, the image of the Yawning class in Figure 8 is incorrectly predicted as a Talking class. The Euclidean distance to the predicted class is proven to be close using Formula 4, which compares the average feature vector distances of correctly and incorrectly predicted images. This makes the image unpredictable because the resulting image vector is close to the average vector of the predicted class.

In this study, comparisons were also made with other methods that also used YawDDataset [20]. The results of this comparison can be seen in Table 10. The LBP boosting accuracy value in Table 10 is obtained from the results of the k-fold cross-validation with k=4. The LBP with the boosting technique gets the best accuracy value compared to the other methods.

Table 10. Comparison of model accuracy values

Method	Acc
MTCNN + Boost LBP + SVM (this work)	**89%**
Variational Descriptor [37]	83%
Yang [27]	83.4%

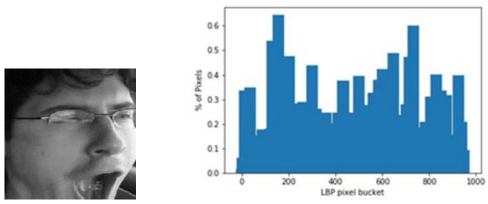


Figure 7. An example of an incorrectly predicted yawning class image as a normal class. The Euclidean distance of the feature vector to the actual class (yawning) is 1.68, while the Euclidean distance to the predicted class (normal) is 1.66.

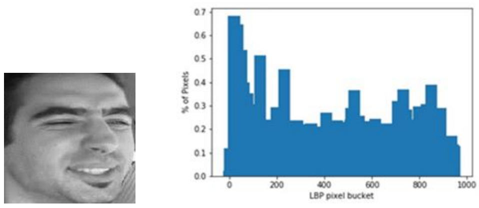


Figure 8. An example of an image of a yawning class that is incorrectly predicted as a talking class. The Euclidean distance of the feature vector to the actual class (yawning) is 1.33, while the Euclidean distance to the predicted class (talking) is 1.32.

4. Conclusion

Based on our findings in this study, fatigue detection in car drivers based on facial images can be well recognized using the Boosting Local Binary Patterns. The proposed system can achieve the best performance with an accuracy value of 93.68% and a precision and recall value of 94%. The best results were achieved on the LBP parameter with R = 2 and P = 4 with an image block size of 20x20. The proposed fatigue detection system outperformed the existing method regarding recognition rate and overall performance. Future research should improve the selection of keyframes to reduce the number of inappropriate class labels included in the training process. Additionally, the use of more data variations will have a significant impact on the system's enhanced ability to detect driver fatigue.

Reference

[1] Kementerian Komunikasi dan Informatika Republik Indonesia, “Rata-rata Tiga Orang Meninggal Setiap Jam Akibat Kecelakaan Jalan,” Aug. 22, 2017. https://www.kominfo.go.id/index.php/content/detail/10368/rata-rata-tiga-orang-meninggal-setiap-jam-akibat-kecelakaan-jalan/0/artikel_gpr (accessed Dec. 20, 2022).

[2] L. H. Tjakranata, B. L. Priatna, and Bianpoen, “Pengaruh Kelelahan Pengemudi terhadap Frekuensi Kecelakaan Lalu Lintas: studi kasus Pengemudi Bus di Jalan Tol Jakarta-Cikampek,” Universitas Indonesia, Jakarta, 1997.

[3] P. Badgujar and P. Selmokar, “Driver gaze tracking and eyes off the road detection,” *Mater Today Proc*, Oct. 2022, doi: 10.1016/J.MATPR.2022.10.046.

[4] W. Li, J. Huang, G. Xie, F. Karray, and R. Li, “A survey on vision-based driver distraction analysis,” *Journal of Systems Architecture*, vol. 121, p. 102319, Dec. 2021, doi: 10.1016/J.SYSARC.2021.102319.

[5] G. Sikander and S. Anwar, “Driver Fatigue Detection Systems: A Review,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 6, pp. 2339–2352, Jun. 2019, doi: 10.1109/TITS.2018.2868499.

[6] M. K. Kamti and R. Iqbal, “Evolution of Driver Fatigue Detection Techniques—A Review From 2007 to 2021,” *Transp Res Rec*, vol. 2676, no. 12, pp. 486–507, Jul. 2022, doi: 10.1177/03611981221096118.

[7] Kementerian Perhubungan Republik Indonesia, “Kelelahan Jadi Faktor Utama Penyebab Kecelakaan,” Aug. 28, 2012. https://dephub.go.id/post/read/kelelahan-jadi-faktor-utama-penyebab-kecelakaan-14470 (accessed Dec. 20, 2022).

[8] M. Marsaid, M. Hidayat, and A. Ahsan, “Faktor yang Berhubungan Dengan Kejadian Kecelakaan Lalu Lintas Pada Pengendara Sepeda Motor di Wilayah Polres Kabupaten Malang,” *Journal of Nursing Science Update (JNSU)*, vol. 1, no. 2, pp. 98–112, Nov. 2013.

[9] M. Ramzan, H. U. Khan, S. M. Awan, A. Ismail, M. Ilyas, and A. Mahmood, “A Survey on State-of-the-Art Drowsiness Detection Techniques,” *IEEE Access*, vol. 7, pp. 61904–61919, 2019, doi: 10.1109/ACCESS.2019.2914373.

[10] Q. Ji and X. Yang, “Real-Time Eye, Gaze, and Face Pose Tracking for Monitoring Driver Vigilance,” *Real-Time Imaging*, vol. 8, no. 5, pp. 357–377, Oct. 2002, doi: 10.1006/RTIM.2002.0279.

[11] J. Zhang, Y. Wu, Y. Chen, J. Wang, J. Huang, and Q. Zhang, “Ubi-Fatigue: Toward Ubiquitous Fatigue Detection via Contactless Sensing,” *IEEE Internet Things J*, vol. 9, no. 15, pp. 14103–14115, Aug. 2022, doi: 10.1109/JIOT.2022.3146942.

[12] A. Kircher, M. Uddman, and J. Sandin, “Vehicle control and drowsiness,” 2002, Accessed: Dec. 20, 2022. [Online]. Available: http://urn.kb.se/resolve?urn=urn:nbn:se:vti:diva-4880

[13] J. Healey and R. Picard, “Smart Car: Detecting driver stress,” *Proceedings - International Conference on Pattern*

DOI: https://doi.org/10.29207/resti.v7i5.4798

Recognition, vol. 15, no. 4, pp. 218–221, 2000, doi: 10.1109/ICPR.2000.902898.

[14] D. Huang, C. Shan, M. Ardabilian, Y. Wang, and L. Chen, “Local binary patterns and its application to facial image analysis: A survey,” *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews*, vol. 41, no. 6, pp. 765–781, Nov. 2011, doi: 10.1109/TSMCC.2011.2118750.

[15] R. Senaratne, B. Jap, S. Lal, A. Hsu, S. Halgamuge, and P. Fischer, “Comparing two video-based techniques for driver fatigue detection: classification versus optical flow approach,” *Machine Vision and Applications 2011 22:4*, vol. 22, no. 4, pp. 597–618, May 2011, doi: 10.1007/S00138-011-0321-4.

[16] W. Liu, H. Sun, and W. Shen, “Driver fatigue detection through pupil detection and yawing analysis,” *ICBBT 2010 - 2010 International Conference on Bioinformatics and Biomedical Technology*, pp. 404–407, 2010, doi: 10.1109/ICBBT.2010.5478931.

[17] D. F. Dinges and R. Grace, “PERCLOS: A Valid Psychophysiological Measure Of Alertness As Assessed By Psychomotor Vigilance,” *Washington, DC : Federal Highway Administration, Office of Motor Carriers, Office of Motor Carrier Research and Standards*, Oct. 1998.

[18] A. al Redhaei, Y. Albadawi, S. Mohamed, and A. Alnoman, “Realtime Driver Drowsiness Detection Using Machine Learning,” *2022 Advances in Science and Engineering Technology International Conferences, ASET 2022*, 2022, doi: 10.1109/ASET53988.2022.9734801.

[19] T. Azim, M. A. Jaffar, M. Ramzan, and A. M. Mirza, “Automatic fatigue detection of drivers through yawning analysis,” *Communications in Computer and Information Science*, vol. 61, pp. 125–132, 2009, doi: 10.1007/978-3-642-10546-3_16/COVER.

[20] S. Abtahi, M. Omidyeganeh, S. Shirmohammadi, and B. Hariri, “YawDD: A yawning detection dataset,” *Proceedings of the 5th ACM Multimedia Systems Conference, MMSys 2014*, pp. 24–28, 2014, doi: 10.1145/2557642.2563678.

[21] S. Abtahi, B. Hariri, and S. Shirmohammadi, “Driver drowsiness monitoring based on yawning detection,” *Conference Record - IEEE Instrumentation and Measurement Technology Conference*, pp. 1606–1610, 2011, doi: 10.1109/IMTC.2011.5944101.

[22] L. Pauly and D. Sankar, “Detection of drowsiness based on HOG features and SVM classifiers,” *Proceedings of 2015 IEEE International Conference on Research in Computational Intelligence and Communication Networks, ICRCICN 2015*, pp. 181–186, Mar. 2016, doi: 10.1109/ICRCICN.2015.7434232.

[23] X. Li, L. Hong, J. chun Wang, and X. Liu, “Fatigue driving detection model based on multi-feature fusion and semi-supervised active learning,” *IET Intelligent Transport Systems*, vol. 13, no. 9, pp. 1401–1409, Sep. 2019, doi: 10.1049/IET-ITS.2018.5590/CITE/REFWORKS.

[24] B. C. Yin, X. Fan, and Y. F. Sun, “Multiscale Dynamic Features Based Driver Fatigue Detection,” *Intern J Pattern Recognit Artif Intell*, vol. 23, no. 3, pp. 575–589, Nov. 2009, doi: 10.1142/S021800140900720X.

[25] Y. Luo, C. M. Wu, and Y. Zhang, “Facial expression recognition based on fusion feature of PCA and LBP with SVM,” *Optik - International Journal for Light and Electron Optics*, vol. 124, no. 17, pp. 2767–2770, Sep. 2013, doi: 10.1016/J.IJLEO.2012.08.040.

[26] C. Chen, H. Seo, C. H. Jun, and Y. Zhao, “Pavement crack detection and classification based on fusion feature of LBP and PCA with SVM,” *International Journal of Pavement Engineering*, vol. 23, no. 9, pp. 3274–3283, 2021, doi: 10.1080/10298436.2021.1888092.

[27] H. Yang, L. Liu, W. Min, X. Yang, and X. Xiong, “Driver Yawning Detection Based on Subtle Facial Action Recognition,” *IEEE Trans Multimedia*, vol. 23, pp. 572–583, 2021, doi: 10.1109/TMM.2020.2985536.

[28] F. Wang, Z. Li, F. He, R. Wang, W. Yu, and F. Nie, “Feature Learning Viewpoint of Adaboost and a New Algorithm,” *IEEE Access*, vol. 7, pp. 149890–149899, 2019, doi: 10.1109/ACCESS.2019.2947359.

[29] C. Xiao, N. Chen, C. Hu, K. Wang, J. Gong, and Z. Chen, “Short and mid-term sea surface temperature prediction using time-series satellite data and LSTM-AdaBoost combination approach,” *Remote Sens Environ*, vol. 233, p. 111358, Nov. 2019, doi: 10.1016/J.RSE.2019.111358.

[30] Z. Mehmood and S. Asghar, “Customizing SVM as a base learner with AdaBoost ensemble to learn from multi-class problems: A hybrid approach AdaBoost-MSVM,” *Knowl Based Syst*, vol. 217, p. 106845, Apr. 2021, doi: 10.1016/J.KNOSYS.2021.106845.

[31] W. Deng and R. Wu, “Real-Time Driver-Drowsiness Detection System Using Facial Features,” *IEEE Access*, vol. 7, pp. 118727–118738, 2019, doi: 10.1109/ACCESS.2019.2936663.

[32] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, “Joint Face Detection and Alignment Using Multitask Cascaded Convolutional Networks,” *IEEE Signal Process Lett*, vol. 23, no. 10, pp. 1499–1503, Oct. 2016, doi: 10.1109/LSP.2016.2603342.

[33] Y. Freund, “Boosting a Weak Learning Algorithm by Majority,” *Inf Comput*, vol. 121, no. 2, pp. 256–285, Sep. 1995, doi: 10.1006/INCO.1995.1136.

[34] Y. Zhang and C. Hua, “Driver fatigue recognition based on facial expression analysis using local binary patterns,” *Optik (Stuttg)*, vol. 126, no. 23, pp. 4501–4505, Dec. 2015, doi: 10.1016/J.IJLEO.2015.08.185.

[35] N. Sun, W. Zheng, C. Sun, C. Zou, and L. Zhao, “Gender classification based on boosting local binary pattern,” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 3972 LNCS, pp. 194–201, 2006, doi: 10.1007/11760023_29/COVER.

[36] P. Yang, S. Shan, W. Gao, S. Z. Li, and D. Zhang, “Face recognition using Ada-Boosted Gabor features,” *Proceedings - Sixth IEEE International Conference on Automatic Face and Gesture Recognition*, pp. 356–361, 2004, doi: 10.1109/AFGR.2004.1301556.

[37] B. Akrout and W. Mahdi, “Yawning detection by the analysis of variational descriptor for monitoring driver drowsiness,” *IPAS 2016 - 2nd International Image Processing, Applications and Systems Conference*, Mar. 2017, doi: 10.1109/IPAS.2016.7880127.

DOI: https://doi.org/10.29207/resti.v7i5.4798