Implementation of Scale-Invariant Feature Transform Convolutional Neural Network for Detecting Distracted Driver

Nahdatul Fhadila[1], Winita Sulandari[2,*], Irwan Susanto[3], Isnandar Slamet[4], Sugiyanto[5], Sri Subanti[6], Etik Zukhronah[7], Hilman Fernandus Pardede[8], Jimmy Abdel Kadar[9]

[1,2,3,4,5,6,7]*Department of Statistics, Universitas Sebelas Maret, Surakarta, Indonesia*

[7,8]*Research Center for Artificial Intelligence and Cyber Security, National Research and Innovation Agency, Bandung, Indonesia*

(Received: May 20, 2024; Revised: June 25, 2024; Accepted: July 01, 2024; Available online: July 16, 2024)

Abstract

A distraction while driving a vehicle may result in fatal consequences, namely accidents that may leave road users seriously injured or even dead. In order to mitigate this risk, it is imperative to establish a distracted driver detection system that is both precise and real-time. This research focuses on the application of artificial intelligence, with a particular emphasis on deep learning, which is achieved through the utilization of the Convolutional Neural Network (CNN) model. In order to enhance the detection of inattentive drivers and produce a more precise model, a scale-invariant feature transform (SIFT)-CNN combination is proposed. The activities of the driver while operating a vehicle are categorized into ten categories in this study. One of these categories is considered a normal condition, while the remaining nine are classified as inattentive behaviors. This study implemented Adam optimization with 64 batches, a learning rate of 0.001, and epochs of 20, 25, 50, and 100. The proposed CNN-SIFT model is capable of achieving superior performance in comparison to the solitary CNN model, as evidenced by the experimental results. The CNN-SIFT model has achieved 99% accuracy and a 0.05 loss when the hyperparameter configuration is optimized for 50 epochs. The analysis indicates that the accuracy of the features obtained from CNN-SIFT can be improved by approximately 1% compared with CNN to classify the type of driver distraction behavior. The model's reliability was further enhanced by its evaluation on test data, which resulted in high accuracy, precision, recall, and F1-score values. The model's ability to accurately identify driver behavior with a high degree of reliability is demonstrated by these results, which are a positive contribution to the improvement of road safety.

Keywords: Distracted Driving Behavior, Convolutional Neural Network, Scale-Invariant Feature Transform

1. Introduction

The Central Bureau of Statistics Indonesia, also known as Badan Pusat Statistik (BPS), reports that the number of transport accidents in Indonesia remains high. In 2022, Indonesia experienced 139,258 traffic accidents, with 20.20% of them culminating in fatalities [1]. Generally, traffic accidents can be caused by a variety of factors, including human, infrastructure, and environmental factor [2], [3]. It is crucial to raise public awareness of the importance of adhering to traffic regulations and maintaining focus while traveling in order to ensure the safety of oneself and others. Certain distractions, such as the use of a cellphone, eating and drinking, listening to the radio, talking to passengers while driving, driving under the influence, and fatigue, may increase the accident statistics, particularly in the context of land transportation [4], [5]. Consequently, in order to resolve this matter, it is necessary to implement a safety road management system [5].

The study on development of driver behaviour detection system started in 2016 when the State Farm Insurance Company USA held a competition on Kaggle to develop a model for detecting distracted drivers using physiological and biomedical sensors such as muscle activity, brain activity, and heart rate. However, the competition failed due to personnel involvement and hardware costs [6]. Various solutions were then found using Support Vector Machine (SVM) models to detect cellphone usage while driving [7]. Recently, many researchers have focused on driver movement detection models that disrupt driving focus due to their role in causing traffic accidents. Elamrani et al. [8] reviewed several references related to the use of various machine learning methods for driving behavior assessment. Some studies, such as [9], [10], [11] have demonstrated that CNN is one of the machine learning methods that

***Corresponding author: Winita Sulandari (winita@mipa.uns.ac.id)**
DOI: https://doi.org/10.47738/jads.v5i3.222

successfully models driver behavior. Masood et.al [9] developed and implemented CNN to detect driver distraction or inattention to support safety road management system. Satardekar [10] used CNN models like VGG-16, VGG-19, and Inception with ensemble techniques to improve accuracy and avoid overfitting. Jagadale & Attar [11] compared AlexNet, VGG-16, and ResNet-50 with their proposed MyModel. The CNN model emerged as a solution for driver detection, and many studies have proven CNN to be the most effective technique for achieving the highest accuracy [12]. The CNN technique is capable of recognizing specific features in images. However, using CNN may have limitations on training data, requiring large amounts of data and computational resources [13].

As time progresses, research on driver behavior has continued to advance. Huang et al. [14] conducted research to detect distracted driving behavior using the Hybrid Framework CNN (HFC), consisting of 3 main modules: the CNN module, feature fusion module, and feature classification module. The CNN module utilized models like Inception v3, ResNet-50, and Xception to extract features, which were then combined with handcrafted HOG features in the feature fusion module, and the feature classification module used fully connected layers. Alkinani et al. [15] used a combination of CNN and handcrafted features in three stages: feature extraction, fusion, and classification. They employed four CNN models to extract features, then merged them with handcrafted HOG features before classification using KNN and SVM methods. Many studies have been conducted on the use of CNN and handcrafted features in analyzing driver behavior.

Handcrafted features are manually created by humans to extract information from data. These features have been widely used in computer vision problems, especially for image classification tasks [16], [17]. They are obtained from non-learning processes by applying various direct operators to image pixels and have the ability to provide several properties, such as rotation and scale invariance [16]. Handcrafted features, which have been the focus of research for many researchers, include several important types, such as Histogram of Oriented Gradients (HOG), Local Binary Patterns (LBP), Speeded-Up Robust Features (SURF), Scale-Invariant Feature Transform (SIFT), and Oriented FAST and Rotated BRIEF (ORB).

The importance of selecting handcrafted features in feature extraction in the classification process has also been extensively studied. Pena [18] compared the performance of SIFT, (SURF), and FAST algorithms in image feature detection. Gupta et al. [19] conducted research on object recognition using SIFT feature detection and (ORB), with the research showing that SIFT features outperformed others. Faturohman et al. [20] conducted a comparison study of SIFT, SURF, and ORB feature detection algorithms in object detection processes in CCTV videos, where the SIFT feature algorithm achieved the highest accuracy of 89.67%. Marlinda et al. [21] conducted a comparative study of the SIFT and ORB methods in identifying Buddha statue faces, with the research showing that ORB produced fewer keypoints than SIFT. Bansal et al. conducted research on object recognition in images using three famous feature descriptor algorithms, namely SIFT, SURF, and ORB, with the research showing that SIFT achieved the highest accuracy [22]. Many studies have proven that SIFT can handle scale, rotation, lighting changes, and can improve high-feature extraction accuracy, making it suitable for use in complex data. However, SIFT also has a drawback in that it does not provide a fixed-length representation of the input image (vector), requiring additional logic for descriptor encoding [23].

In recent years, combining the advantages of SIFT with CNN has attracted increasing interest [24]. Most studies suggest combining SIFT and CNN features before the classification stage [25], [26]. Bousaid et al. [13] combined SIFT features with CNN features to recognize facial expressions in images, achieving an accuracy improvement of 99.35%. Tyagi and Bansal [27] combined the feature from accelerated segment test (FAST) and SIFT approaches with Indian sign language hand gesture recognition automatically and accurately, achieving CNN and SIFT accuracies of 97.89%. Tsourounis et al. [23] developed the CNN-SIFT approach, combining the strength of handcrafted SIFT features with CNN, emphasizing the importance of image representation in data-driven systems.

This study discusses the use of handcrafted features combined with deep learning architecture. The purpose of the combination is to enhance the ability of handcrafted features to provide information to CNN and enhance CNN with rotation, complex textures, and patterns. This study combines local features from CNN and handcrafted features such as SIFT to improve the system's ability to recognize complex objects and visual patterns. In the context of driver behavior problems involving complex images, the CNN-SIFT framework is proposed for further research on these

images. The CNN model focuses on spatial features, while SIFT is reliable in features that are invariant to scale and rotation shifts. The proposed CNN-SIFT model has higher efficiency compared to CNN trained directly on images. By combining CNN-SIFT, which has feature extraction capabilities of CNN models and local rotation invariant features of SIFT, local scale and rotation invariance can improve accuracy and FI-score in driver behaviour recognition. Combining CNN-SIFT can enhance model performance and complement the weaknesses of each model.

2. Literature and Research Framework

2.1. Convolutional Neural Network

CNN is a type of neural network used for processing images [28]. The architecture of CNN consists of multiple layers, and each layer learns to generate increasingly complex features as the network progresses [28], [29]. The advantages of CNN include its ability to produce relevant features from images, weight sharing to save time and computational memory, and its applicability to various tasks such as classification, segmentation, object recognition, image enhancement, and transfer learning [30], [31], [32], [33], [34]. CNNs consist of two types of layers: feature extraction layers consisting of convolutional layers, ReLU activation functions, and pooling layers, and classification layers consisting of fully-connected layers and softmax activation functions [35]. An example of CNN architecture is shown in figure 1.

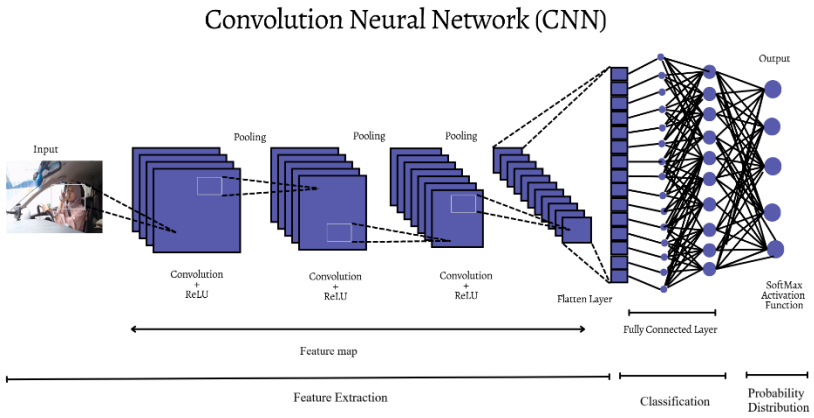


Figure 1. CNN architecture

2.2. Scale Invariant Feature Transform (SIFT)

SIFT is a computer vision algorithm developed by David Lowe and published in 1999 [16]. SIFT inherently detects features that are stable to changes in scale and rotation. Consequently, the results of SIFT descriptors will not be substantially influenced by many geometric transformations that are frequently employed in image augmentation. Additionally, SIFT features are generally quite resilient to minor adjustments in perspective and position, rendering additional augmentations less advantageous. The process of extracting SIFT features entails the identification of key points and the computation of descriptors surrounding those key points. This procedure has optimized the detection of informative and dependable features in a variety of circumstances [16], [36].

In this study, SIFT algorithm aims to extract key features (keypoints) from images that are robust to changes in scale, rotation, and illumination. The main stages of SIFT include detecting scale-space extrema, identifying significant keypoints, determining orientation for each keypoint, and constructing keypoint descriptors to represent the local image structure. This approach allows SIFT to extract consistent and reliable features from images for object recognition applications.

2.3. Confusion Matrix

Model evaluation is an important step to assess the effectiveness of a program or training. This evaluation process utilizes various metrics and techniques that are suitable for the goals and types of programs being evaluated. The main evaluation parameters include accuracy, recall, precision, and F1-score for each class. These values are calculated using

a confusion matrix that includes true positive (TP), true negative (TN), false positive (FP), and false negative (FN) [37].

$$\text{accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \quad (1)$$

$$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (2)$$

$$\text{recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (3)$$

$$\text{F1} - \text{score} = 2 \times \frac{\text{recall} \times \text{precision}}{\text{recall} + \text{precision}} \quad (4)$$

Accuracy measures the level of agreement between predictions and actual values. Precision and recall measure the model's performance in classification, with F1-score as the harmonic average of both, indicating the balance between precision and recall. F1-score is useful when false positives and false negatives have serious consequences and as a reference for the classification model's performance.

2.4. The Proposed Method

This research utilizes a CNN architecture by combining handcrafted SIFT features. The research process involves three key steps. First step is feature extraction from images using both CNN and SIFT models. Second step includes extracted features combination from CNN and SIFT models and the last step is classification and validation. In this step, the new image input (outsample) is test to detect whether the driver's behavior is distracted or not.

Figure 2 shows the CNN-SIFT architecture used in this study. The research employs ReLU and softmax activation functions. The ReLU activation function is used to learn more complex feature representations from the input data. This function produces a value of 0 if the input is negative and the input value itself if the input is positive [38]. The ReLU function is widely popular and effective in speeding up the model training process while reducing the risk of overfitting [39]. Softmax is an activation function at the output that produces probabilities for each output class. Softmax is used to classify inputs into various classes based on the probabilities generated [40].

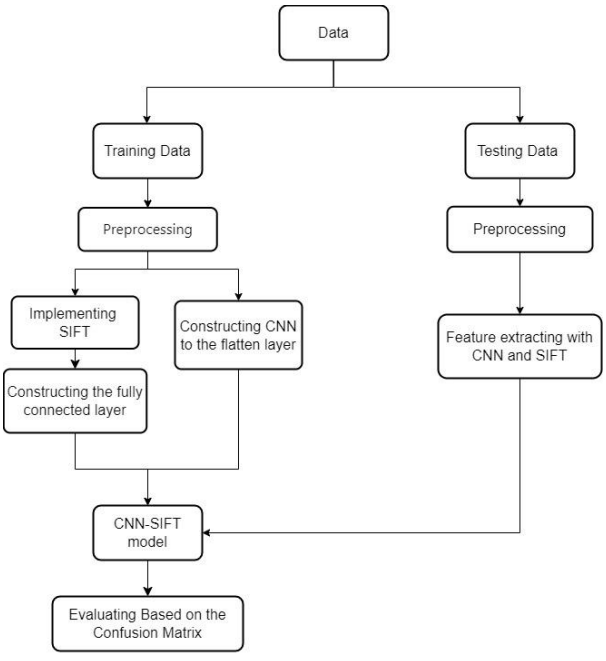


Figure 2. Illustration of Model Incorporation in Research

3. Research Methodology

3.1. Experimental Setup

This research utilizes the "Four-Wheeler Driver Behavior Images" dataset from the National Research and Innovation Agency. This dataset involves four participants demonstrating ten different driver behaviors. The dataset acquisition process involved using a camera positioned on the left side of the car dashboard.

Table 1. Dataset Description

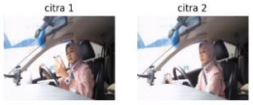
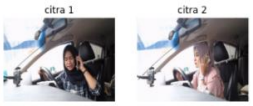
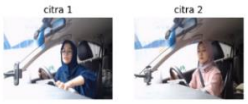
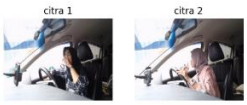
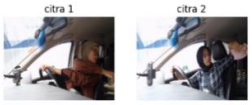
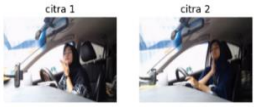
Class	Description
C0	Normal driving
C1	Texting with right hand
C2	Talking on the phone with right hand
C3	Texting with left hand
C4	Talking on the phone wiht left hand
C5	Operating the radio
C6	Dringking
C7	Reaching behind
C8	Grooming
C9	Talking with passengers

In the experimental study, all computations were performed using a notebook with the following specifications: device name LAPTOP-F054KD9L, AMD Ryzen 5 5500U processor with Radeon Graphics at 2.10 GHz, 16.0 GB installed RAM (15.3 GB usable), and a 64-bit operating system with a google Collaboratory (google colab).

During the recording process, videos were captured at a resolution of 640x480 pixels. The dataset used consists of 3000 images, with 300 images per class, totaling 10 classes. The dataset is grouped into two categories of driver behavior, namely safe and distracted. Each class of data is labeled according to the driver's behavior, as listed in table 1. Drivers exhibiting safe behavior are placed in class C0, while drivers with distracted behavior are placed in classes C1-C9. Table 2 shows sample examples from each class of driver images.

Table 2. Example Samples of Driver Behavior Images for Each Class

No	Driver Behavior	Example
1	Normal driving	
2	Texting with right hand	
3	Talking on the phone with right hand	

4	Texting with left hand	
5	Talking on the phone wiht left hand	
6	Operating the radio	
7	Dringking	
8	Reaching behind	
9	Grooming	
10	Talking with passengers	

This data was processed using Google Colaboratory with a T4 GPU. Before modeling, the data used in this research was divided into 80% training data and 20% testing data. Data preprocessing involved several steps used to prepare and clean raw data before analysis or model creation. Dataset preprocessing techniques in this research include resize and rescale. The resize step involves changing the image size from 640x480 resolution to 256x256 pixels. Resizing the images provides several benefits, including more efficient use of computational resources due to smaller size, consistency in dataset size for easier data processing and model implementation, and reducing the risk of overfitting. The next step is image rescaling, which is done by dividing each pixel in the image by the value 255. The purpose of this step is to change the range of pixel intensities to be between 0 and 1.

This research conducted modeling using ADAM optimization, which is an optimization method used to update parameters using a learning rate of 0.001. The categorical cross-entropy loss function is used to measure how close the predicted model probability distribution is to the actual target probability. The training process was conducted with a batch size of 64, determining the number of samples processed in one iteration, and using epochs of 20, 25, 50, and 100.

3.2. CNN Model

In this modeling, a CNN model is applied using RGB color image inputs. The convolutional model for RGB color image inputs is constructed using several layers. Starting from the input layer with a shape of (256, 256, 3), representing an image with three channels (RGB), followed by a convolutional layer (Conv2D) with 32 filters, a kernel size of (3, 3), padding 'same', and ReLU activation function. This process is continued with a MaxPooling layer (MaxPool2D) with a pool size of (2, 2) to reduce the dimensions of the image. These convolutional and pooling layers are repeated to create more complex feature representations. Afterward, the image is flattened (Flatten) and continued with a Dense layer with 100 neurons and ReLU activation. The last Dense layer has 10 neurons (corresponding to the number of

ISSN 2723-6471

classes) with softmax activation function to produce classification output. Table 3 represents the construction of the CNN model along with its parameters. The final results consist of accuracy and loss values for both training and testing to provide an overview of how well the model can improve image classification. Table 4 presents the accuracy and loss values using the CNN model.

Table 3. Architecture of the CNN model

Step	Layer	Shape	Parameter
1	InputLayer	(256,256,3)	0
2	Conv2D	(256,256,32)	896
3	max_pooling2d	(128,128,32)	0
4	Conv2D	(128,128,64)	18.496
5	max_pooling2d	(64,64,64)	0
6	flatten	(262144)	0
7	dense	(100)	26.214.500
8	dense	(10)	1.010
	Total params		26.234.902
	Trainable params		26.234.902
	Non-trainable params		0

Table 4. Accuracy and Loss obtained from the training and testing data for the CNN Model

Epoch	Training		Testing	
	Accuracy	Loss	Accuracy	Loss
20	100%	$1{,}31\times10^{-4}$	98,33%	0,07
25	100%	$1{,}51\times10^{-4}$	98,33%	0,05
50	100%	$1{,}28\times10^{-5}$	98,17%	0,08
100	100%	$7{,}50\times10^{-7}$	98,20%	0,11

The CNN model in table 3 shows fluctuations in performance. At epoch 20, the accuracy reaches 98.33%, but the loss value is relatively high at 0.07. At epoch 25, there is an accuracy of 98.33%, accompanied by a lower loss value of 0.05. By epoch 50, there is a decrease in accuracy to 98.17%, and the loss value increases to 0.08. At epoch 100, the accuracy increases again to 98.20%, but with a corresponding increase in the loss value to 0.11. Table 5 shows the performance results of the best CNN model epoch 25.

Table 5. Performance of each class for the best CNN Model

Class	Precision (%)	Recall (%)	F1-score (%)
C0	100	95	97
C1	100	97	98
C2	95	100	98
C3	99	94	96
C4	94	100	97
C5	98	98	98
C6	100	100	100

C7	100	100	100
C8	100	100	100
C9	97	98	98
average	**98**	**98**	**98**
accuracy		**98**	

3.3. CNN-SIFT Model

In this model, a combined model is employed consisting of two parts: the first part processes RGB image data using a CNN model, and the second part processes image data whose features are extracted using the SIFT algorithm.

In the first part, or the RGB input part, it starts with an input layer sized (256, 256, 3) representing the image with three color channels. Then, a convolution operation is performed with a Conv2D layer having 32 filters, kernel size (3, 3), padding 'same', and using ReLU activation function. Subsequently, pooling is carried out with a MaxPool2D layer with a pool size of (2, 2). This process is repeated with the next Conv2D layer having 64 filters and concluded with a Flatten layer.

The second part, or the SIFT Input part, begins with an input layer sized (128,) according to the dimensions of the extracted SIFT features. This operation is then flattened. This combined model involves a Dense layer with 128 neurons and linear activation for the SIFT input. Then, using a Concatenate layer, the outputs from the RGB and SIFT parts are merged. The process continues with a Dense layer having 100 neurons and ReLU activation function, and an output Dense layer with 10 neurons corresponding to the number of classes and using softmax activation. Table 6 represents the construction of the CNN-SIFT model along with its parameters.

The final results consist of the accuracy and loss values for both training and testing, providing an overview of how well the model can improve image classification. Table 7 shows the accuracy and loss values using the combined SIFT and CNN model.

Table 6. Architecture of the CNN-SIFT Model

Step	layer	Shape	Parameter
1	InputLayer RGB	(256,256,3)	0
2	Conv2D RGB	(256,256,32)	896
3	max_pooling2d RGB	(128,128,32)	0
4	Conv2D RGB	(128,128,64)	18.496
5	max_pooling2d RGB	(64,64,64)	0
6	Flatten RGB	(262144)	0
7	InputLayer SIFT	(128)	0
8	Flatten SIFT	(128)	0
9	dense SIFT	(128)	16.512
10	concatenate	(262272)	0
11	Dense	(100)	26.227.300
12	Dense	(10)	1010
	Total params		**26.264.214**
	Trainable params		**26.264.214**
	Non-trainable params		**0**

 ISSN 2723-6471

Table 7. Accuracy and Loss obtained from the training and testing data for the CNN-SIFT Model

Epoch	Training		Testing	
	Accuracy	Loss	Accuracy	Loss
20	100%	$4{,}67\times10^{-4}$	99,00%	0,06
25	100%	$5{,}19\times10^{-4}$	98,66%	0,05
50	100%	$3{,}44\times10^{-5}$	99,00%	0,05
100	100%	$1{,}46\times10^{-6}$	98,66%	0,08

The combined SIFT and CNN model in table 4 shows that at epoch 50, the accuracy reaches 99.00% with a loss value of 0.06. Meanwhile, at epoch 25, despite a slight decrease in accuracy to 98.66%, the loss value remains low at 0.05. At epoch 50, the accuracy increases again to 99.00% with a loss of 0.05, and at epoch 100, the accuracy remains high at 98.66% with a loss of 0.08. table 8 shows the performance results of the best CNN-SIFT model.

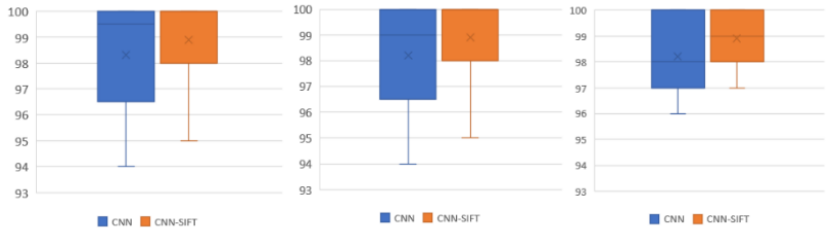
Table 8. Performance of each class for the best CNN-SIFT Model

Class	Precision (%)	Recall (%)	F1-score (%)
C0	100	100	100
C1	100	98	99
C2	98	100	99
C3	100	95	97
C4	95	100	98
C5	98	98	98
C6	100	100	100
C7	100	100	100
C8	100	100	100
C9	98	98	98
average	**99**	**99**	**99**
accuracy		**99**	

Overall, the CNN-SIFT model in table 7 demonstrates more consistent performance with high accuracy levels and low loss values at each epoch stage. In contrast, the CNN model in table 4 exhibits fluctuating performance with varying accuracy and loss values. Therefore, the use of the CNN-SIFT model can be considered a more stable and effective approach in image classification compared to a single CNN model.

3.4. Discussion

This study do not provide any statistical significancy testing to determine the differences between CNN and CNN-SIFT. However, the Boxplot comparison of precision, recall, and F1-score in figure 3 illustrates that CNN-SIFT produces a narrower range of values with a higher median than those obtained from CNN model. This demonstrates that the CNN-SIFT model outperforms the CNN model.



(a) Precision (b) recall (c) F1-score

Figure 3. Boxplot comparison for (a) Precision, (b) Recall and (c) F1-score obtained from the best CNN and CNN-SIFT model

In case of running the CNN model with 25 epochs, it consumes 130.69 seconds for the training time and needs 3,766.48 MB of memory. In contrast, CNN-SIFT necessitates a memory of 3,198.72 MB and a training duration of 237.09 seconds. CNN-SIFT training duration is nearly twice as long as CNN due to the additional steps involved in the SIFT process. It is intriguing that CNN-SIFT necessitates less memory to generate more precise class predictions than CNN models. The results are consistent for other optimal epochs.

Based on the experimental study, both accuracy and loss show the consistent patterns for the training data, where the accuracies increase and the losses decrease as the number of epochs increases. Meanwhile, the testing data exhibit fluctuations that are believed to be the consequence of overfitting, which may be precipitated by the use of insufficiently representative data during the training process (see table 4 and table 6). A more thorough examination is required to investigate the factors that may contribute to the fluctuation results in the accuracy and loss of testing data. In practical applications, further study is needed to address the challenge of overfitting. The network needs to be trained with more diverse data that can represent all possible driver behavior patterns.

For the real-world deployment, The CNN-SIFT approach can be applied through the development of an application program for mobile phones that combines the camera sensor capabilities of the mobile phone with a CNN-SIFT based detection program. The camera on the mobile phone which is located close to the driver will record the driver's behaviors while driving, so that when it is discovered that behaviors endanger security and safety, the application will immediately provide a warning to the driver or passengers in the vehicle. The warning can be in the form of a sound or text notification that can be received directly or transferred via Bluetooth to other mobile phones. However, this system certainly requires precise and sensitive camera sensor capabilities for capturing images, a relatively large mobile phone memory capacity, and adequate mobile phone processor speed.

Besides that, further research into the development of combining CNN with other handcrafted feature approaches is an intriguing area to explore. HOG and SURF are two other handcrafted feature approaches that are worth studying in combination with CNN. The HOG has similarities with SIFT in that it similarly extracts an area surrounding a key point and constructs a histogram of gradients from this region. The primary distinction lies in the fact that the area is larger and the cells are configured to overlap. Hence, the HOG contributes to capturing more precise information. Meanwhile, the SURF utilizes an integral image, sometimes referred to as a summed-area table, which is a computational method and data structure for rapidly and effectively calculating the total sum of values inside a rectangular segment of a grid. It demonstrates superior performance compared to other descriptors, such as SIFT, in a consistent and substantial manner [12]. Therefore, the application of the hybrid CNN-HOG and CNN-SURF schemes for driver behavior detection is interesting for further research.

4. Conclusion

This study illustrates significant efforts in enhancing the detection of distracted driving behavior through artificial intelligence approaches, particularly by leveraging a CNN model combined with handcrafted features using SIFT. The experiments demonstrate that the combined SIFT and CNN model provides more stable and effective performance compared to a single CNN model, particularly achieving the highest accuracy of 99.00% at epoch 50. Although the single CNN model exhibits greater fluctuations with decreases in accuracy and increases in loss values at certain epoch

stages. These fluctuations can be attributed to insufficiently representative data, inadequate dataset size and quality, and the complexity of the number of layers used.

The research findings suggest that the combined SIFT and CNN approach can be considered a more reliable solution for improving image classification related to driver behavior. Therefore, for future research, it is recommended to optimize model parameters and structures to achieve better results and gain a deeper understanding of the feature interactions from both input sources. It is also suggested to explore handcrafted features further and develop hybrid models to deepen the understanding of feature interactions from various input sources. Additionally, diversifying data and integrating the model into smart vehicle technology can enhance the generalization and practical applicability of this research.

Subsequent research should involve studying driver behavior data obtained directly from the field to achieve more relevant results in real-world situations. Furthermore, addressing the challenges and requirements for real-world implementation, such as reliability, integration with existing systems, scalability, security, and compliance with industry regulations, will provide a clearer path for the practical use of this model in smart vehicles.

5. Declaration

5.1. Author Contributions

Conceptualization: N.F., W.S., I.S., I.S., S., S.S., E.Z., H.F.P., and J.A.K.; Methodology: W.S., I.S., S., S.S., and E.Z.; Software: N.F. and H.F.P.; Validation: N.F., W.S., I.S., I.S., S., S.S., E.Z., H.F.P., and J.A.K.; Formal Analysis: N.F., W.S., I.S., I.S., S., S.S., E.Z., H.F.P., and J.A.K.; Investigation: N.F.; Resources: W.S., I.S., S., S.S., and E.Z.; Data Curation: E.Z.; Writing Original Draft Preparation: N.F., W.S., I.S., I.S., S., S.S., E.Z., H.F.P., and J.A.K.; Writing Review and Editing: E.Z., H.F.P., N.F., W.S., I.S., I.S., S., S.S., and J.A.K.; Visualization: N.F. and H.F.P.; All authors have read and agreed to the published version of the manuscript.

5.2. Data Availability Statement

The data presented in this study are available on request from the corresponding author.

5.3. Funding

We thank LPPM UNS for supporting this research through Research Group Grant No. 194.2/UN27.22/PT.01.03/2024.

5.4. Institutional Review Board Statement

Not applicable.

5.5. Informed Consent Statement

Not applicable.

5.6. Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] Badan Pusat Statistik (BPS), "Jumlah Kecelakaan, Korban Mati, Luka Berat, Luka Ringan, dan Kerugian Materi," 2023. [Dataset]. Available: https://www.bps.go.id/id/statistics-table/2/NTEzIzI=/jumlah-kecelakaan--korban-mati--luka-berat--luka-ringan--dan-kerugian-materi.html. [Accessed: 18-Jun-2024].

[2] H. R. Zadry, A. B. Cahyono, and S. Utomo, "Traffic Accident in Indonesia and Blind Spot Detection Technology—An Overview," *in Conference Proceedings of HUMENS (International Human Engineering Symposium,*vol.2021, no. 10, pp. 231–242. doi: 10.1007/978-981-16-4115-2_18.

[3] A. D. Saputra, "Studi Tingkat Kecelakaan Lalu Lintas Jalan di Indonesia Berdasarkan Data KNKT (Komite Nasional Keselamatan Transportasi) dari Tahun 2007-2016," *Warta Penelitian Perhubungan,* vol. 29, no. 2, p. 179-182, Jul. 2018, doi: 10.25104/warlit.v29i2.557.

[4] R C. Irwin, S. Monement, and B. Desbrow, "The Influence of Drinking, Texting, and Eating on Simulated Driving Performance," *Traffic Injury Prevention*, vol. 16, no. 2, pp. 116–123, 2014. doi: 10.1080/15389588.2014.920953.

[5] K. Srinivasan, L. Garg, , D. Datta, A.A. Alaboudi, N.Z. Jhanjhi, R. Agarwal, and A.G. Thomas,, "Performance Comparison of Deep CNN Models for Detecting Driver's Distraction," *Computers, Materials & Continua*, vol. 68, no. 3, pp. 4109–4124, 2021, doi: 10.32604/cmc.2021.016736.

[6] A. Fernández, R. Usamentiaga, J. L. Carús, and R. Casado, "Driver distraction using visual-based sensors and algorithms," *Sensors (Switzerland)*, vol. 16, no. 11, pp. 1–44, 2016, doi: 10.3390/s16111805.

[7] D. Tran, H.M. Do, W. Sheng, H. Bai, G. Chowdhary. "Real-time Detection of Distracted Driving Based on Deep Learning," *IET Intelligent Transport Systems*, vol 12, no. 10, pp. 1210-1219. doi: 10.1049/iet-its.2018.5172

[8] Z. E.A. Elassad, H. Mousannif, H. Al Moatassime, and A. Karkouch, "The application of machine learning techniques for driving behavior analysis: A conceptual framework and a systematic literature review," *Eng Appl Artif Intell*, vol. 87, no. 1, pp. 1-18. 2020, doi: 10.1016/j.engappai.2019.103312.

[9] S. Masood, A. Rai, A. Aggarwal, M. N. Doja, and M. Ahmad, "Detecting distraction of drivers using Convolutional Neural Network," *Pattern Recognit Lett*, vol. 139, no. 1, pp. 79–85, 2020, doi: 10.1016/j.patrec.2017.12.023.

[10] P.M. Chawan, S. Satardekar, D. Shah, R.Badugu, and, A. Pawar, "Distracted Driver Detection and Classification", *Int. Journal of Engineering Research and Application*, vol. 8, no. 4 (Part-III), pp.60-64., April 2018, doi: 10.9790/9622-0804036064.

[11] T. Huang and R. Fu, "Driver Distraction Detection Based on the True Driver's Focus of Attention," *in IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 10, pp. 19374-19386, Oct. 2022, doi: 10.1109/TITS.2022.3166208.

[12] M. D. Hssayeni, S. Saxena, R. Ptucha, and A. Savakis, "Distracted driver detection: Deep learning vs handcrafted features," *IS and T International Symposium on Electronic Imaging Science and Technology*, vol. 10, no 1, pp. 20–26, 2017, doi: 10.2352/ISSN.2470-1173.2017.10.IMAWM-162.

[13] R. Bousaid, M. El Hajji, and Y. Es-Saady, "Facial Expression Recognition Using a Hybrid ViT-CNN Aggregator," *Internasional Journal of Computer Vision*, vol 130, no.4, pp. 61–70, 2022, doi: 10.1007/978-3-031-06458-6_5.

[14] C. Huang, X. Wang, J. Cao, S. Wang, and Y. Zhang, "HCF: A Hybrid CNN Framework for Behavior Detection of Distracted Drivers," *IEEE Access*, vol. 8, no. 1, pp. 109335–109349, 2020, doi: 10.1109/ACCESS.2020.3001159.

[15] M. H. Alkinani, W. Z. Khan, Q. Arshad, and M. Raza, "HSDDD: A Hybrid Scheme for the Detection of Distracted Driving through Fusion of Deep Learning and Handcrafted Features," *Sensors*, vol. 22, no. 5, pp. 123-135, 2022, doi: 10.3390/s22051864.

[16] D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *Int J Comput Vis*, vol. 60, no. 2, pp. 91–110, Nov. 2004, doi: 10.1023/B:VISI.0000029664.99615.94.

[17] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-Up Robust Features (SURF)," *Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346-359, 2008, doi: 10.1016/j.cviu.2007.09.014.

[18] A. Chater and A. Lasfar, " New Approach to the Identification of the Easy Expression Recognition System by Robust Techniques (SIFT, PCA-SIFT, and SURF)," *TELKOMNIKA (Telecommunication Computing Electronics and Control*, vol 18, no.2, pp. 695-704, 2020. doi: 10,12928/telkomnika.v18i2.13726

[19] S. Gupta, M. Kumar, and A. Garg, "Improved object recognition results using SIFT and ORB feature detector," *Multimed Tools Appl*, vol. 78, no. 23, pp. 34157–34171, 2019, doi: 10.1007/s11042-019-08232-6.

[20] O. Yakovlena and K. Nikolaieva,"Research of Descriptor Based Image Normalization and Comparative Analysis of SURF, SIFT, BRISK, ORB, KAZE, AKAZE Descriptors," *Advanced Information Systems*, vol. 4 , no. 4, pp. 89-101, 2021. doi: 10.20998/2522-9052.2020.4.13

[21] P. Chhabra, N.K. Garg, and M. Kumar,"Content-based Image Retrieval System Using ORB and SIFT Features, " *Neural Comput & Applic*, vol.32. No.8, pp.2725-2733, 2020. doi: 10.1007/s00521-018-3677-9.

[22] M. Bansal, M. Kumar, and M. Kumar, "2D object recognition: a comparative analysis of SIFT, SURF and ORB feature descriptors," *Multimedia Tools and Applications*, vol. 80, no. 12, pp. 18839–18857, May 2021, doi: 10.1007/s11042-021-10646-0.

[23] D. Tsourounis, D. Kastaniotis, C. Theoharatos, A. Kazantzidis, and G. Economou, "SIFT-CNN: When Convolutional Neural Networks Meet Dense SIFT Descriptors for Image and Sequence Classification," *J Imaging*, vol. 8, no. 10, pp. 1-18, Sep. 2022, doi: 10.3390/jimaging8100256.

[24] L. Zheng, Y. Yang, and Q. Tian, "SIFT Meets CNN: A Decade Survey of Instance Retrieval," *IEEE Trans Pattern Anal Mach Intell*, vol. 40, no. 5, pp. 1224–1244, May 2018, doi: 10.1109/TPAMI.2017.2709749.

[25] W. Lin, K. Hasenstab, G. Moura Cunha, and A. Schwartzman, "Comparison of handcrafted features and convolutional neural networks for liver MR image adequacy assessment," *Sci Rep*, vol. 10, no. 1, pp. 20336-20350, Nov. 2020, doi: 10.1038/s41598-020-77264-y.

[26] H. Wang and S. Hou, "Facial Expression Recognition based on The Fusion of CNN and SIFT Features," *in Proc of 2020 IEEE 10th International Conference on Electronics Information and Emergency Communication (ICEIEC), 17-19 July 2020, Beijing, China*, vol 2020, no. July, pp. 190–194. doi: 10.1109/ICEIEC49280.2020.9152361.

[27] A. Tyagi and S. Bansal, "Hybrid FiST_CNN approach for Feature Extraction for Vision-Based Indian Sign Language Recognition," *The International Arab Journal of Information Technology*, vol. 19, no. 3, pp.403-411, 2022, doi: 10.34028/iajit/19/3/15.

[28] W. Rawat and Z. Wang, "Deep Convolutional Neural Networks for Image Classification: A Comprehensive Review," *Neural Computation*, vol. 29, no. 9, pp. 1-98, 2017, doi: 10.1162/NECO_a_00990.

[29] X. Shi, F. Lv, D. Seng, J. Zhang, J. Chen, and B. Xing, " Visualizing and Understanding Graph Convolutional Network," *Multimedia Tools and Applications*, vol.80, no. November, pp. 8355-8375, 2021. doi: 10.1007/s11042-020-09885-4.

[30] C. Lam, C. Yu, L. Huang, and D. Rubin, "Retinal Lesion Detection with Deep Learning Using Image Patches, " *Investigate Ophthalmology & Visual Science*, vol. 59, no. 1, pp. 590-596, 2018. Doi: 10.1167/iovs.17-22721.

[31] M. Melinsca, P. Prentasic, and S. Loncaric, "Retinal vessel segmentation using deep neural networks," *VISAPP 2015 - 10th International Conference on Computer Vision Theory and Applications; VISIGRAPP, Proceedings*, vol. 1, no. June 2018, pp. 577–582, 2015, doi: 10.5220/0005313005770582.

[32] S. Kido, Y. Hirano and N. Hashimoto, "Detection and classification of lung abnormalities by use of convolutional neural network (CNN) and regions with CNN features (R-CNN)," *2018 International Workshop on Advanced Image Technology (IWAIT), Chiang Mai, Thailand*, vol. 2018, no. 5, pp. 1-4, doi: 10.1109/IWAIT.2018.8369798.

[33] B. Qin, J. Qian, Y. Xin, B. Liu, and Y. Dong, " Distracted driver Detection Based on a CNN with Decreasing Filter Size," *IEEE Transactions on Intelligent Transportation System*, vol 23, no.7, pp.6922-6933, 2021. Doi: 10.1109/TITS.2021.3063521.

[34] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans Knowl Data Eng*, vol. 22, no. 10, pp. 1345–1359, 2010, doi: 10.1109/TKDE.2009.191.

[35] A. L. Katole, K. P. Yellapragada, A. K. Bedi, S. S. Kalra, and M. Siva Chaitanya, "Hierarchical Deep Learning Architecture for 10K Objects Classification," *In Computer Science & Information Technology Conference Proceedings*, vol. 5, no. 14, pp. 77–93, 2015, doi: 10.5121/csit.2015.51408.

[36] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," *IEEE Trans Pattern Anal Mach Intell*, vol. 27, no. 10, pp. 1615–1630, 2005, doi: 10.1109/TPAMI.2005.188.

[37] K. Lan, D. Wang, S. Fong, L. Liu, K. Wong, and N. Dey, "A Survey of Data Mining and Deep Learning in Bioinformatics," *Journal of Medical Systems*, vol. 42, no. 139, pp. 1-20, 2018, doi: 10.1007/s10916-018-1003-9.

[38] W.B. Langdon, and S.M. Guatafson, " Genetic Programming and Evolvable Machines: ten years of reviews," *Genetic Programming and Evolvable Machines*, vol 11, no. September, pp. 321-338, 2010. doi: 10.1007/s10710-010-9111-4

[39] Y. Bai, "RELU-Function and Derived Function Review," *In SHS Web of Conferences: Proc. of International Conference on Science and Technology Etnics and Human Future (STEHF 2022)*, vol. 144, no. 2022, pp. 1–5, 2022. doi: 10.1051/shsconf/202214402006.

[40] S. Sharma, S. Sharma, and A. Athaiya, "Activation functions in neural networks," *International Journal of Engineering Applied Sciences and Technology*, vol. 4, no. 12, pp. 310-316, Apr. 2020.