Journal of Mechatronics, Electrical Power, and Vehicular Technology 14 (2023) 177-185

Journal of Mechatronics, Electrical Power, and Vehicular Technology

e-ISSN: 2088-6985
p-ISSN: 2087-3379

mev.brin.go.id

# Distracted driver behavior recognition using modified capsule networks

Jimmy Abdel Kadar [a, *], Margareta Aprilia Kusuma Dewi [b], Endang Suryawati [a], Ana Heryana [a], Vicky Zilfan [a], Budiarianto Suryo Kusumo [a, c], Raden Sandra Yuwana [a], Ahmad Afif Supianto [a, d], Hasih Pratiwi [b], Hilman Ferdinandus Pardede [a]

[a] *Research Center for Artificial Intelligence and Cyber Security, National Research and Innovation Agency*
*Kawasan Sains dan Teknologi (KST) Samaun Samadikun, Jalan Sangkuriang, Bandung, 40135, Indonesia*

[b] *Faculty of Mathematics and Natural Sciences, Sebelas Maret University*
*Ir. Sutami Street No.36A, Surakarta, 57126, Indonesia*

[c] *Faculty of Electrical Engineering and Information Technology, Chemnitz University*
*Technische Universität Chemnitz, Straße der Nationen 62, D-09111 Chemnitz, Germany*

[d] *Department of ICT and Natural Sciences, Norwegian University of Science and Technology*
*Larsgårdsvegen 2, Ålesund, 6009, Norway*

Received 20 September 2023; 1st Revision 28 November 2023; 2nd Revision 4 December 2023;
Accepted 5 December 2023; Published online 29 December 2023

## Abstract

Human activity recognition (HAR) is an increasingly active study field within the computer vision community. In HAR, driver behavior can be detected to ensure safe travel. Detect driver behaviors using a capsule network with leave-one-subject-out validation. The study was done using CapsNet with leave-one-subject-out validation to identify driving habits. The proposed method in this study consists of two parts, namely, encoder and decoder. The encoder used in this study modifies Sabour's capsule network architecture by adding a convolution layer before going to the primary capsule layer. The proposed method is evaluated using a primary dataset with 10 classes and 300 images for each class. The dataset is split based on hold-out validation and leave-one-subject-out validation. The resulting models were then compared to conventional CNN architecture. The objective of the research is to identify driving behavior. In this study, the proposed method results in an accuracy rate of 97.83 % in the split dataset using hold-out validation. However, the accuracy decreased by 53.11 % when the proposed method was used on a split dataset using leave-one-subject-out validation. This is because the proposed method extracts all features, including the attributes of each participant contained in the input image (user-independent). Thus, the resulting model in this study tends to overfit.



Keywords: capsule network; driver behavior detection; human activity recognition.

## I. Introduction

The computer vision community's interest in human activity recognition (HAR) is growing due to the need to construct intelligent systems such as monitoring, control, and analysis [1][2]. The primary objective of HAR is to determine and predict what humans do based on a set of information [3]. One implementation of HAR is for the recognition of activity during driving.

A vision-based technique can be used to detect driving behavior [4][5]. The driver's head, torso, upper arms, lower arms, and hands may be captured using a camera mounted on the car's dashboard. The categorization of distracted driving behaviors is typically the focus of the analysis of driving behaviors. This category might alert other drivers to their circumstances, lowering the chance of a collision and ensuring a safe journey [6].

Convolutional neural network (CNN) is a deep learning algorithm that is a leading method to address this problem [7]. Example [8], a conventional CNN architecture with three

* Corresponding Author. Tel: +62-811897211
*E-mail address*: jimmy.abdel.kadar@brin.go.id

doi: https://dx.doi.org/10.14203/j.mev.2023.v14.177-185
2088-6985 / 2087-3379 ©2023 National Research and Innovation Agency

MEV is Scopus indexed Journal and accredited as Sinta 1 Journal (https://sinta.kemdikbud.go.id/journals/detail?id=814)
How to Cite: J.A. Kadar *et al.*, "Distracted driver behavior recognition using modified capsule networks," *Journal of Mechatronics, Electrical Power, and Vehicular Technology*, vol. 14, no. 2, pp. 177-185, Dec. 2023.

convolutional layers, three pooling layers, and three fully-connected layers was used to classify driving behaviors based on side-view photographs.

C. Yan *et al.* [9] classified driving behaviors based on front-view and side-view pictures with their optical fluxes by combining two stream inputs with interwoven CNN. K. A. AlShalfan *et al.* [10] classified drivers' behavior based on side-view photos using modified VGG-16. X. Rao [11] driving behaviors are identified based on side view photographs using PCA whitening pre-processing and typical CNN architecture, including four convolutional layers, four pooling layers, and two fully-connected layers.

The majority of studies that have used CNN for driver behavior detection have shown excellent results. However, there might still be a few problems. To begin with, CNN is not equivalent to an affine transformation [12]. Additionally, spatial information in picture data can be removed by downsampling on the pooling layer [13][14]. However, CNN is not equivariant to affine transformation [15][16]. By employing capsules rather than neurons and a dynamic routing method to retain the spatial associations between features, a capsule network (CapsNet) can be utilized to address these shortcomings [17][18][19].

CapsNet architecture has been used widely for image classification. CapsNet architecture was able to recognize handwritten Indic characters [20], Devanagari manuscript [21], Car dataset, and Solar panel dataset [22]. Some studies have also modified CapsNet architecture. F. Kinli *et al.* [23] showed that CapsNet was modified by adding three more convolution layers to detect the Fashion dataset. G. Madhu *et al.* [24] showed that CapsNet was modified by adding four more convolution layers to detect the existence of a malaria parasite in a cell. Fire recognition has become crucial, in area safety using CapsNet [25].

The hold-out validation approach is typically used in image classification to randomly divide the data into training and validation sets. However, for HAR, other sources provided the data. If those training and validation sets were randomly divided, the model might view data from the same subject throughout training and validation [26]. The model's generalizability to new users (user-independent) suffers due to this method. As a result, the split data approach known as leave-one-subject-out is employed.

The objective of the research is to identify driving behavior. The study was done using CapsNet with leave-one-subject-out validation to identify driving habits. This work adds a convolution layer to Sabour's CapsNet architecture to deepen the model before moving on to the top capsule layer. The datasets used to divide by hold-out and leave-one-subject-out validation, the model built from this architecture is expected to deliver great generalization and superior performance than the conventional CNN design.

## II. Materials and Methods

### A. Capsule network

A capsule serves as the primary low-level node of a type of neural network known as a "CapsNet" [27]. Vectors "length and orientation represent the entities" existence and attributes in the vector activation functions used by CapsNet. CapsNets's utility in resolving challenging computer vision issues has grown with recent developments in their routing methods [28]. CapsNet stores data at a vector level instead of convolutional neural networks [29].

The amplitude and direction of the vector neuron in CapsNet are identical to those of an average vector [30]. The length of the vector neuron represents the probability of an object being present at a particular position in the image. Meanwhile, the orientation represents the image's attitude.

A capsule in the layer contains an activity vector used to estimate the instantiation parameters of the secondary capsule at the layer using the trainable weight matrix, as shown in equation (1). The prediction vector shows the contribution made by the first capsule to the secondary capsule. In CapsNet, a dynamic routing algorithm maintains spatial relationships between features. Dynamic routing between capsules was first developed by Sara Sabour. It is used to train the CapsNet iteratively. Each primary capsule in the bottom layer $l$ delivers all capsules in the subsequent layer $l+1$. The matrix transformation will then anticipate the secondary capsule's instantiation parameters. The result of the matrix transformation represents the agreement with the secondary capsule. If the multiple predictions agree, then the two capsules are relevant to each other, and it will activate the secondary capsule [31].

A capsule $i$ in the layer $l$ contains an activity vector $u_i$ that is used to estimate the instantiation parameters $\hat{u}_{j|i}$ of the subsequent capsule $j$ at layer $l+1$ applying the trainable weight matrix $w_{ij}$, as shown in equation (1). The prediction vector $\hat{u}_{j|i}$ reflects how much the central capsule $i$ assists the subsequent capsule $j$.

$$\hat{u}_{i|j} = w_{ij} u_j \tag{1}$$

A coupling parameter $c_{ij}$ connected with the prediction vector indicates the agreement between both capsules. The coupling coefficient $c_{ij}$ of capsule $i$ is determined by routing softmax, which can be seen in equation (2), representing $b_{ij}$ the log prior probability that the capsule $i$ is associated with the capsule $j$, beside that $k$ is number of classes and $\exp(b_{ik})$ is exponential for output vector.

$$c_{ij} = \frac{\exp(b_{ij})}{\sum_k \exp(b_{ik})} \tag{2}$$

Equation (3) calculates a weighted total of $s_j$ of all these main capsule predictions, which is the output of the secondary capsule , and $x_{ij}$ are coupling coefficients that are determined by the iterative dynamic routing process.

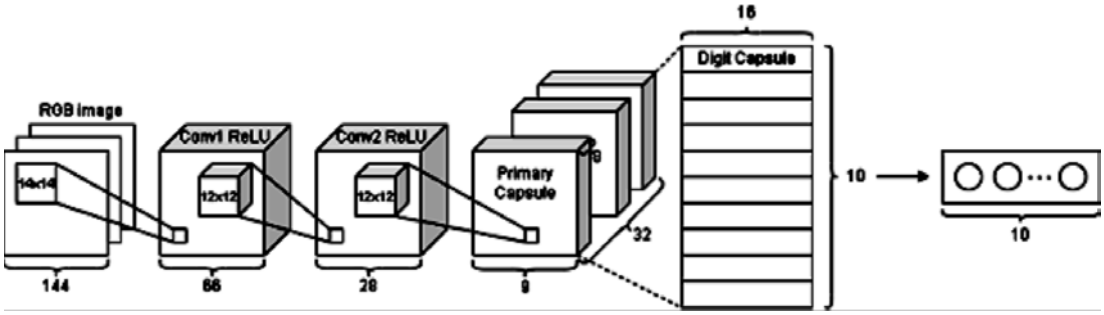


Figure 1. The encoder architectures

$$s_j = \sum_i x_{ij} \cdot \hat{u}_{j|i} \quad (3)$$

The resultant output is then squashed using the activation function $v_j$ and make sure that the length of the capsule result is between 0 and 1. Equation (4) depicts the squashing activation function.

$$v_j = \frac{\|s_j\|^2}{1+\|s_j\|^2} \cdot \frac{s_j}{\|s_j\|} \quad (4)$$

As shown in equation (5), the agreement between the expected and actual outputs is computed by calculating their dot product. The capsules create an exact spatial connection if the resultant dot product is a large scalar ($a_{ij}$).

$$a_{ij} = v_j \cdot \hat{u}_{j|i} \quad (5)$$

B. Margin loss

Margin loss ($L_k$) is mathematically defined as in equation (6).

$$L_k = T_k \max(0, m^+ - \|v_k\|)^2 + \lambda(1 - T_k) \max(0, \|v_k\| - m^-)^2 \quad (6)$$

where $T_k = 1$ for the correct prediction and $T_k = 0$ otherwise. $v_k$ = vector obtained from DigitCaps layer. The higher $m^+ = 0.9$ and the lower $m^- = 0.1$ thresholds for the correct and wrong classes, respectively. Meanwhile, $\lambda = 0.5$ is employed for numerical stability.

C. The architecture

An auto-encoder is embedded into the architecture. A decoder and an encoder are two essential elements. The encoder structure, as seen in Figure 1, is the initial component. It includes four layers, including a fully linked digit capsule layer and three convolutional layers. Following the ReLU activation function, the first convolutional layer consists of 128 kernels, each measuring 10 × 10 units, and it operates with a stride of 2. The ReLU activation function is followed by the second convolution layer, which has 256 8 × 8 kernels with a stride of 2. In 2D pictures, the first two convolutional layers identify fundamental characteristics.

The third layer is a convolutional capsule layer representing the primary capsule layer. It contains 8D convolutional capsules with 32 channels. Each essential capsule employs eight convolutional units with a kernel of 8 × 8 and a stride of two. The main capsule has an 8D vector with 6 × 6 × 32 capsule outputs. The last layer is the digit capsule layer. It has one 16D capsule for each digit class, and each is fully linked to all the capsules in the preceding layer. A dynamic routing mechanism is used between the main and digit capsule layers.

The second part is the decoder structure, shown in Figure 2. The decoder structure recreates a picture from the output of the proper digit capsule by delivering it into three fully connected layers. These layers learn to recreate a 96 × 96 RGB image by keeping essential features. The loss function is calculated during training by minimizing the Euclidean distance between the reconstructed and input images. The overall CapsNet architecture in detail can be seen in Table 1.

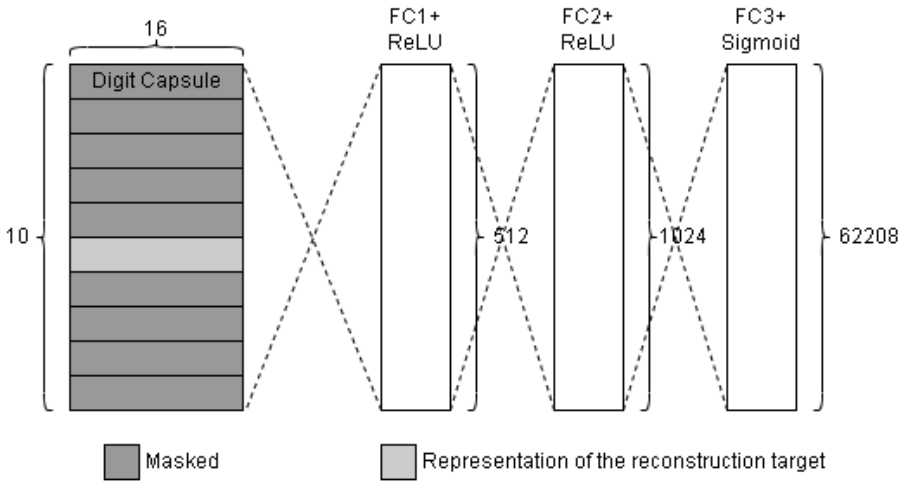


Figure 2. The decoder architectures

Table 1.
The architecture of the proposed method in detail

| Layer | Output Shape | Unit |
|---|---|---|
| Input image | 144, 144, 3 | 0 |
| Convolutional #1 | 66, 66, 128 | 75,392 |
| Convolutional #2 | 28, 28, 256 | 4,718,848 |
| Primary capsule | 9, 9, 32, 8 | 9,437,440 |
| Digit capsule | 16, 10 | 3,369,600 |
| Fully connected #1 | 512 | 82,432 |
| Fully connected #2 | 1024 | 525,312 |
| Fully connected #3 | 62208 | 63,763,200 |
| Total trainable params | | 81,972,224 |

HAR refers to the motion of one or more human bodily parts [32]. HAR aims to automatically interpret human body gestures or motions and determine what human does through a collection of observations [33]. HAR should designate the same action with the same name even if it is performed by different persons in various settings or environments [34]. In this study, two conditions are used for both implementation and evaluation.

The first employs hold-out validation, based on dividing the dataset into two subsets: training and validation sets. 80 % of a dataset is used for training and 20 % for validation. It is less costly to compute because it only has to be performed once, but the model's conclusions may alter if the data is divided again. Hold-out validation indicates that accuracy depends on the subject chosen for evaluation [35].

The second one uses the leave-one-subject-out validation. Leave-one-subject-out validation is a variant of hold-out validation, where one subject is considered for the validation and others for training the model [36]. This approach makes the model evaluate new subjects. Here, we want to observe the model's capability for user independence conditions.

## D. Experimental setup

This study collected 3000 images from four participants performing ten activities in the car. The data from those participants are collected using a Logitech camera placed on the left side of the dashboard. Each participant is asked to perform ten different activities and then recorded. The behavior reflects one safe driving behavior that leads to safe travel and nine behaviors that lead to hazardous travel. Each behavior becomes a classification class in this study, as shown in Table 2, whereas the data samples are shown in Figure 3. The camera position is placed parallel to the subject which can display all the classes described in Table 2 that were tested.

Table 2.
Dataset description

| Class | Description |
|---|---|
| C0 | Safe driving |
| C1 | Texting with right hand |
| C2 | Talking on phone with right hand |
| C3 | Texting with left hand |
| C4 | Talking on phone with left hand |
| C5 | Adjusting radio |
| C6 | Drinking |
| C7 | Reaching behind |
| C8 | Doing hair and makeup |
| C9 | Safe driving |

Assess user-dependent and user-independent models by modeling and evaluating image data through hold-out and leave-one-subject-out validation techniques. In the first approach, the dataset is partitioned into two subsets: the training and validation sets, with a random split of 80 % for training and 20 % for validation. Conversely, three subjects are utilized for the training set in the leave-one-subject-out validation, while one is reserved for the validation set.

The data were rescaled into 144 × 144 pixels for pre-processing, and RGB features were extracted as input and normalized by dividing each pixel in the image by 255 so that each pixel in the data ranges from 0 to 1. Normalizing is done to simplify further calculations.

Google Collab with a standard GPU is used to compile the model. The data is trained with the Adam optimizer and a learning rate of 0.0001. In this research, 100 epochs with a batch size of 60 were utilized to evaluate the proposed method's performance in hold-out validation, and a batch size of 75 was used to evaluate the proposed method's performance in leave-one-subject-out validation.

The suggested method's performance on the model is then assessed using an accuracy measure and a loss function generated from the total margin and reconstruction losses. The model's performance


Figure 3. The samples of the data

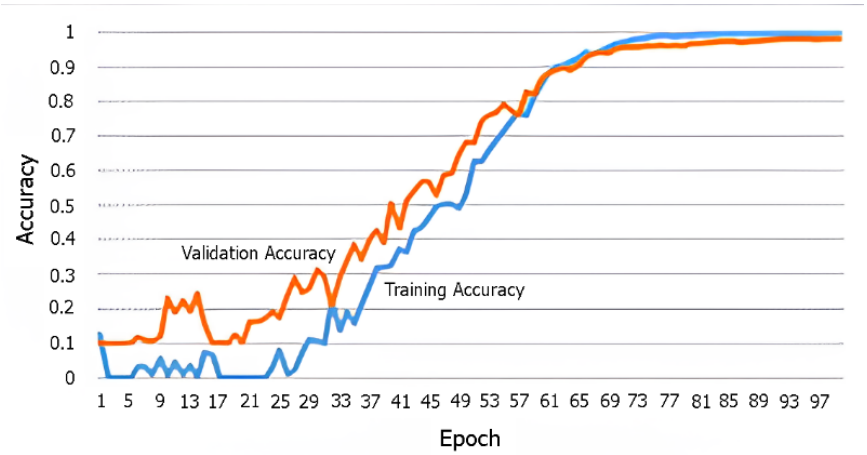


Figure 4. The proposed method's training and validation accuracy are based on hold-out validation

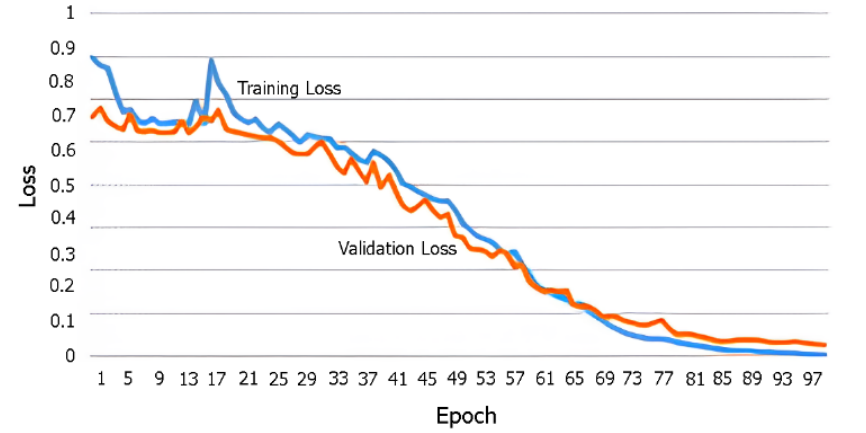


Figure 5. Training and validation loss of the proposed method based on hold-out validation

is then compared to a popular CNN design, which contains three convolutional layers, a pooling layer, and three fully linked layers at the end. The kernel and configuration used in this architecture are the same as those used in CapsNet architecture.

Every subject in the picture has unique information; the convolutionally (CNN) model evaluates each image, regardless of whether the same individual has long or short hair, wearing a headscarf/hijab or not, and so forth. CapsNet is used in conjunction with user-dependent and user-independent models in this study. The hypothesis posits that dependent users, regardless of whether they wear the hijab or not, will yield good accuracy since all subjects are included in the testing population, while independent users will create poor accuracy due to the existence of subjects outside the testing population.

## III. Results and Discussions

### A. Performance of the proposed method based on hold-out validation

Figure 4 and Figure 5 show the accuracy and loss of the suggested technique. The figures show that the suggested approach becomes convergent after the 60$^{th}$ epoch. Furthermore, the difference between the training and validation sets is slight in the 100$^{th}$ epoch. These minor variations demonstrate that the suggested strategy works effectively when all individuals are included in the training and validation sets.

Figure 6 shows the suggested method's confusion matrix. The proposed method can properly and efficiently recognize image data in the behavior of "drinking," "C6," and "reaching behind" "C7". The proposed method may readily distinguish picture data in the behavior of "talking on the phone with the right hand" "C2" and "talking to a passenger" "C9". However, they also retrieve some image data

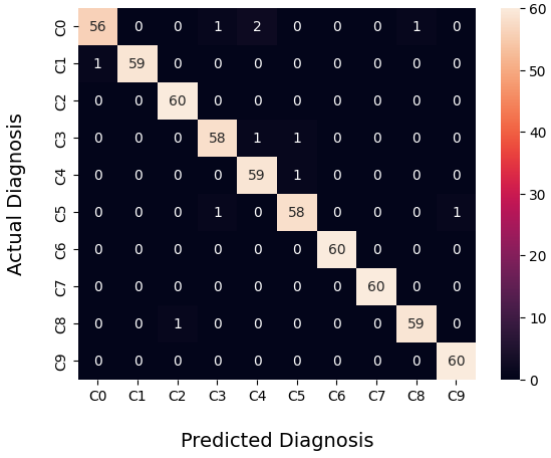


Figure 6. Confusion matrix of the proposed method based on hold-out validation

Table 3.
Performance comparison of the proposed technique with popular CNN architecture based on hold-out validation

| Architecture | Training Set | | Validation Set | |
|---|---|---|---|---|
| | Accuracy (%) | Loss | Accuracy (%) | Loss |
| Proposed method | 100 | 0.0034 | 98.17 | 0.0262 |
| Conventional CNN | 100 | 2.5e-4 | 97.83 | 0.0938 |

from other behaviors that are not relevant to these behaviors. On the other hand, the proposed method can retrieve all the relevant image data in the behavior of "texting with the right hand" "C1", but there is an instance from this behavior that is misclassified to the other behavior. These errors occur due to the similarity of features with other behaviors. This similarity results in low inter-class variability and leads to misclassification.

The proposed method incorporates the reconstruction loss, which calculates the disparity between the input and reconstructed images. The reconstructed images are utilized as regularization to prevent overfitting. Figure 7 shows examples of the reconstructed picture data used in this research.

The proposed method is compared to the popular CNN design, as shown in Table 3. It is used to assess the effectiveness of the suggested approach based on hold-out validation. The loss performance of the popular CNN architecture is roughly 3.58 times greater than the proposed approach. This difference demonstrates that the suggested technique is more likely to predict a value than the current CNN method. As a result, the proposed method works better than the conventional CNN architecture when applied to hold-out validation.

## B. Performance of the proposed method based on leave-one-subject-out validation

The accuracy and loss of the proposed method can be seen respectively in Figure 8 and Figure 9. From those figures, the gap between the training and validation sets is quite prominent in the 100$^{th}$ epoch. This prominent gap shows that the proposed method has not worked well when a new participant is used in the validation set. Nonetheless, the proposed method is still trying to study the features of new participants. It can be seen from the loss graph, which is still decreasing overall. As a result, the proposed approach may recognize the driver behavior of a new participant.

Figure 10 shows the proposed method's confusion matrix. The proposed method can recognize image data in the behavior of "reaching behind" "C7" quite well. However, it retrieves some image data from other behaviors that are not relevant to these behaviors.

However, the proposed method can retrieve the most relevant image data in "talking on the phone

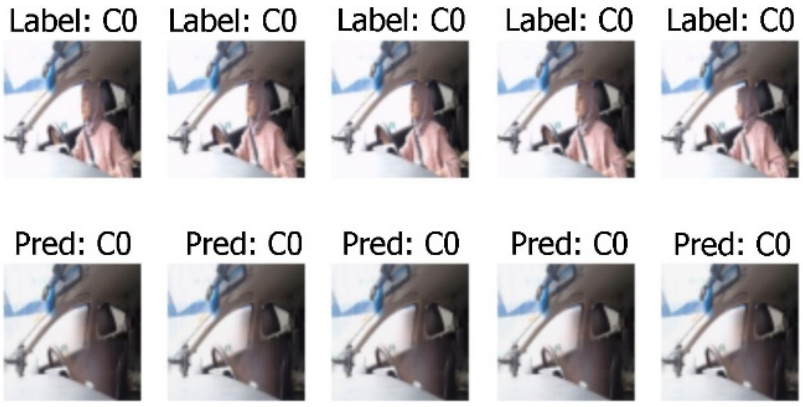


Figure 7. Reconstructed image of the proposed method based on hold-out validation

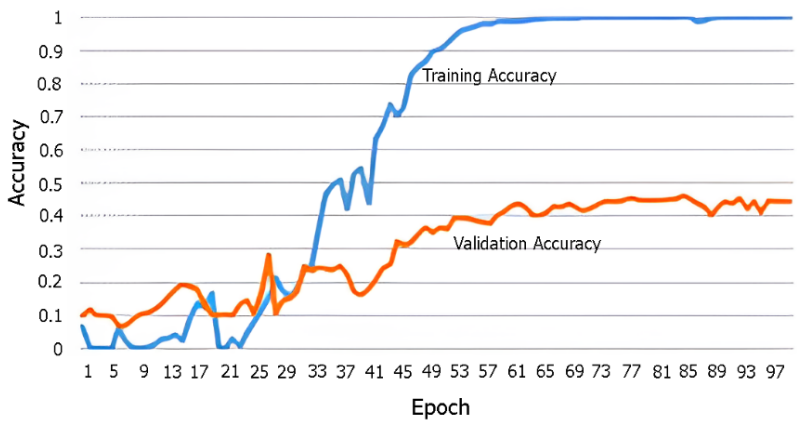


Figure 8. Training and validation accuracy of the proposed method based on leave-one-subject-out validation

with the right hand" and "C2". However, some image data from this behavior is misclassified to the other behavior. These errors occur because a new user (user-independent) is used as an input, which results in a change in intra-class variability so that the proposed method experiences a decrease in performance. The samples of the reconstructed image data in this study can be seen in Figure 11. From that figure, the reconstructed images show that the proposed method has generalized the user. However, the proposed method still extracts unnecessary features and instead loses essential features that make the proposed method unable to distinguish one driver's behavior from another.

The proposed method is then compared to the conventional CNN architecture that can be seen in Table 4. It is used to examine the efficacy of the proposed method employing leave-one-subject-out validation. According to the table, the proposed method outperforms the popular CNN design.

However, the proposed method's effectiveness could be improved in the validation set. In the training set, the proposed approach detects driving behavior effectively. However, it is less able to detect

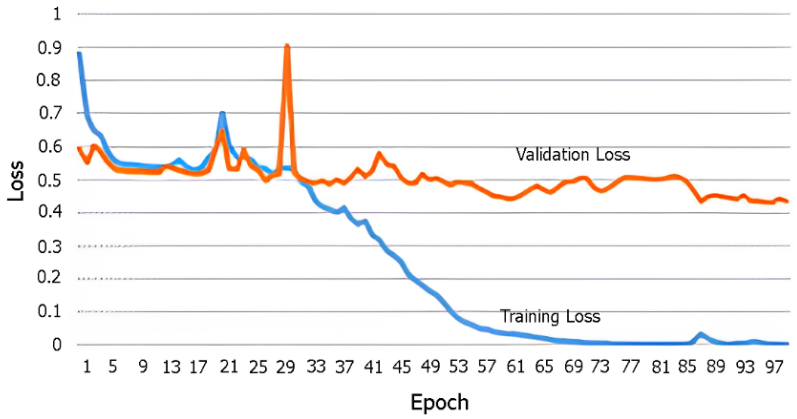


Figure 9. Training and validation loss of the proposed method based on leave-one-subject-out validation

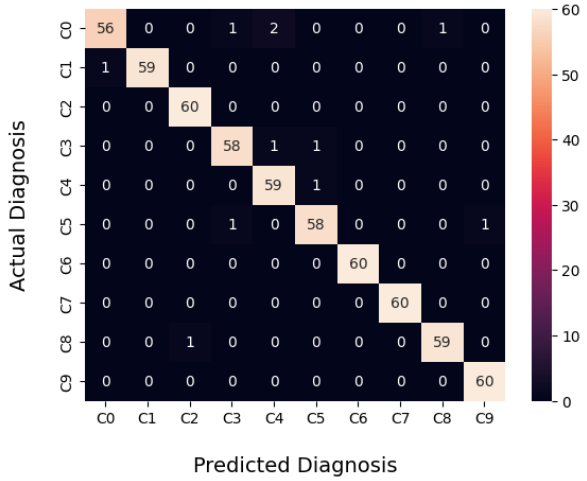


Figure 10. Confusion matrix of the proposed approach based on leave-one-subject-out validation

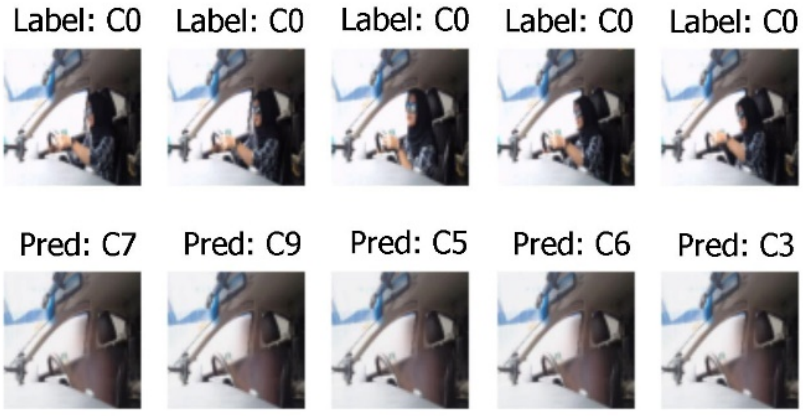


Figure 11. Reconstructed image of the proposed method based on leave-one-subject-out validation

Table 4.
Performance comparison of the proposed method with conventional CNN architecture according to validation with one subject left out

| Architecture | Training Set | | Validation Set | |
|---|---|---|---|---|
| | Accuracy (%) | Loss | Accuracy (%) | Loss |
| Proposed method | 100 | 0.0058 | 44.80 | 0.4310 |
| Conventional CNN | 100 | 4.6e-6 | 36.93 | 8.2506 |

the driver behavior in the validation set due to new participants used in the validation set. In this case, the auto-encoder, as a dimensionality reduction, can still not generate a model with a good generalization.

## IV. Conclusion

In this study, modified Sabour's CapsNet is used to identify the driver behavior. The dataset is modeled using CapsNet architecture. It is then evaluated by using hold-out validation and leave-one-subject-out validation. It is also compared to the conventional CNN architecture to evaluate the effectiveness of the proposed method. The proposed method can provide better performance compared to the conventional CNN when it is applied to hold-out validation because it uses an auto-encoder to avoid overfitting problems. However, the proposed method experienced a decrease in performance by 54.36 % when the new user (user-independent) was used as an input to identify the driver's behavior. This shows that the regularization used in the proposed method is still not robust to user variability. That makes the resulting model still prone to overfitting. Therefore, further study can be performed with better regularization so the resulting performance will remain stable under various circumstances or environments.

## Declarations

### Author contribution

J.A. Kadar and M.A.K. Dewi, contributed equally as the main contributor of this paper. All authors read and approved the final paper.

### Funding statement

This research was funded by Rumah Program Kendaraan Listrik - Research Organization Electronics and Informatics (OREI) - National Research and Innovation Agency and the Faculty of Mathematics and Natural Sciences at Sebelas Maret University. It offers research with knowledge, insight, and expertise.

### 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.

### Additional information

Reprints and permission: information is available at https://mev.brin.go.id/.

Publisher's Note: National Research and Innovation Agency (BRIN) remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

## References

[1] D. Wu, N. Sharma and M. Blumenstein, "Recent advances in video-based human action recognition using deep learning: A review," *2017 International Joint Conference on Neural Networks (IJCNN)*, Anchorage, AK, USA, pp. 2865-2872, 2017.

[2] F. Demrozi, G. Pravadelli, A. Bihorac, and P. Rashidi, "Human Activity Recognition Using Inertial, Physiological and Environmental Sensors: A Comprehensive Survey," *IEEE Access*, vol. 8, pp. 210816–210836, 2020.

[3] M. M. Islam, S. Nooruddin, F. Karray, and G. Muhammad, "Human activity recognition using tools of convolutional neural networks: A state of the art review, data sets, challenges, and future prospects," *Comput Biol Med*, vol. 149, p. 106060, 2022.

[4] S. Zhang, Z. Wei, J. Nie, L. Huang, S. Wang, and Z. Li, "A Review on Human Activity Recognition Using Vision-Based Method," *Journal of Healthcare Engineering*, vol. 2017. Hindawi Limited, 2017.

[5] L. Guarda, J. E. Tapia, E. L. Droguett, and M. Ramos, "A novel Capsule Neural Network based model for drowsiness detection using electroencephalography signals," *Expert Syst Appl*, vol. 201, p. 116977, 2022.

[6] C. Jobanputra, J. Bavishi, and N. Doshi, "Human activity recognition: A survey," in *Procedia Computer Science*, Elsevier B.V., 2019, pp. 698–703.

[7] S. Indolia, A. K. Goswami, S. P. Mishra, and P. Asopa, "Conceptual Understanding of Convolutional Neural Network- A Deep Learning Approach," in *Procedia Computer Science*, Elsevier B.V., 2018, pp. 679–688.

[8] C. Yan, F. Coenen, and B. Zhang, "Driving posture recognition by convolutional neural networks," *IET Computer Vision*, vol. 10, no. 2, pp. 103–114, Mar. 2016.

[9] C. Zhang, R. Li, W. Kim, D. Yoon, and P. Patras, "Driver behavior recognition via interwoven deep convolutional neural nets with multi-stream inputs," *IEEE Access*, vol. 8, pp. 191138–191151, 2020.

[10] K. A. AlShalfan and M. Zakariah, "Detecting Driver Distraction Using Deep-Learning Approach," *Computers, Materials and Continua*, vol. 68, no. 1, pp. 689–704, Mar. 2021.

[11] X. Rao, F. Lin, Z. Chen, and J. Zhao, "Distracted driving recognition method based on deep convolutional neural network," *J Ambient Intell Humaniz Comput*, vol. 12, no. 1, pp. 193–200, Jan. 2021.

[12] V. Sarveshwaran, I. T. Joseph, M. M, and K. P, "Investigation on Human Activity Recognition using Deep Learning," *Procedia Comput Sci*, vol. 204, pp. 73–80, 2022.

[13] N. Akhtar and U. Ragavendran, "Interpretation of intelligence in CNN-pooling processes: a methodological survey," *Neural Computing and Applications*, vol. 32, no. 3. Springer, pp. 879–898, Feb. 01, 2020.

[14] R. Shi and L. Niu, "A brief survey on capsule network," in *Proceedings - 2020 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology, WI-IAT 2020*, Institute of Electrical and Electronics Engineers Inc., Dec. 2020, pp. 682–686.

[15] M. Sun, Z. Song, X. Jiang, J. Pan, and Y. Pang, "Learning Pooling for Convolutional Neural Network," *Neurocomputing*, vol. 224, pp. 96–104, Feb. 2017.

[16] Z. Sun, G. Zhao, R. Scherer, W. Wei, and M. Woźniak, "Overview of Capsule Neural Networks," *Journal of Internet Technology*, vol. 23, no. 1. Taiwan Academic Network Management Committee, pp. 33–44, 2022.

[17] G. E. Hinton, A. Krizhevsky, and S. D. Wang, "Transforming Auto-Encoders," in *Artificial Neural Networks and Machine Learning – ICANN 2011*, W. and G. M. and K. S. Honkela Timo and Duch, Ed., Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 44–51.

[18] S. Sabour, N. Frosst, and G. E. Hinton, "Dynamic Routing Between Capsules," *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pp. 3859–3869, Oct. 2017.

[19] J. Cai, S. Wang, and W. Guo, "Unsupervised embedded feature learning for deep clustering with stacked sparse auto-encoder," *Expert Syst Appl*, vol. 186, p. 115729, 2021.

[20] B. Mandal, S. Dubey, S. Ghosh, R. Sarkhel, and N. Das, "Handwritten Indic Character Recognition using Capsule Networks," in *2018 IEEE Applied Signal Processing Conference (ASPCON)*, 2018, pp. 304–308.

[21] A. Moudgil, S. Singh, V. Gautam, S. Rani, and S. H. Shah, "Handwritten devanagari manuscript characters recognition

using capsnet," *International Journal of Cognitive Computing in Engineering*, vol. 4, pp. 47–54, 2023.
[22] M. L. Mekhalfi, M. B. Bejiga, D. Soresina, F. Melgani, and B. Demir, "Capsule networks for object detection in UAV imagery," *Remote Sens (Basel)*, vol. 11, no. 14, 2019.
[23] F. KINLI and F. KIRAÇ, "FashionCapsNet: Clothing Classification with Capsule Networks," *Bilişim Teknolojileri Dergisi*, vol. 13, no. 1, pp. 87–96, Jan. 2020.
[24] G. Madhu, A. Govardhan, B. S. Srinivas, S. A. Patel, B. Rohit, and B. L. Bharadwaj, "Capsule Networks for Malaria Parasite Classification: An Application Oriented Model," in *2020 IEEE International Conference for Innovation in Technology, INOCON 2020*, Institute of Electrical and Electronics Engineers Inc., Nov. 2020.
[25] Y. Wu, L. Cen, S. Kan, and Y. Xie, "Multi-layer capsule network with joint dynamic routing for fire recognition," *Image Vis Comput*, vol. 139, p. 104825, 2023.
[26] I. Brishtel, S. Krauss, M. Chamseddine, J. R. Rambach, and D. Stricker, "Driving Activity Recognition Using UWB Radar and Deep Neural Networks," *Sensors*, vol. 23, no. 2, Jan. 2023.
[27] E. Juralewicz and U. Markowska-Kaczmar, "Capsule Network Versus Convolutional Neural Network in Image Classification: Comparative Analysis," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Springer Science and Business Media Deutschland GmbH, 2021, pp. 17–30.
[28] S. Choudhary, S. Saurav, R. Saini, and S. Singh, "Capsule Networks for Computer Vision Applications: A Comprehensive Review," *Applied Intelligence*, vol. 53, no. 19, pp. 21799–21826, Jun. 2023.
[29] E. Goceri, *Analysis of Capsule Networks for Image Classification*. International Conference Scientific Computing, 2021.
[30] M. K. Patrick, A. F. Adekoya, A. A. Mighty, and B. Y. Edward, "Capsule Networks – A survey," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 1. King Saud bin Abdulaziz University, pp. 1295–1310, Jan. 01, 2022.
[31] S. J. Pawan and J. Rajan, "Capsule networks for image classification: A review," *Neurocomputing*, vol. 509. Elsevier B.V., pp. 102–120, Oct. 14, 2022.
[32] F. Abdul Manaf and S. Singh, "Computer vision-based survey on Human Activity Recognition system, challenges and applications," in *2021 3rd International Conference on Signal Processing and Communication, ICPSC 2021*, Institute of Electrical and Electronics Engineers Inc., May 2021, pp. 110–114.
[33] O. C. Ann and L. B. Theng, "Human activity recognition: A review," in *2014 IEEE International Conference on Control System, Computing and Engineering (ICCSCE 2014)*, 2014, pp. 389–393.
[34] D. R. Beddiar, B. Nini, M. Sabokrou, and A. Hadid, "Vision-based human activity recognition: a survey," *Multimed Tools Appl*, vol. 79, no. 41–42, pp. 30509–30555, Nov. 2020.
[35] H. Bragança, J. G. Colonna, H. A. B. F. Oliveira, and E. Souto, "How Validation Methodology Influences Human Activity Recognition Mobile Systems," *Sensors*, vol. 22, no. 6, Mar. 2022.
[36] D. Gholamiangonabadi, N. Kiselov, and K. Grolinger, "Deep Neural Networks for Human Activity Recognition with Wearable Sensors: Leave-One-Subject-Out Cross-Validation for Model Selection," *IEEE Access*, vol. 8, pp. 133982–133994, 2020.