An Approach to Hyperparameter Tuning in Transfer Learning for Driver Drowsiness Detection Based on Bayesian Optimization and Random Search

Hoang-Tu Vo, Hoang Tran Ngoc, Luyl-Da Quach
Software Engineering Department
FPT University, Cantho City, Vietnam

***Abstract*—Driver drowsiness is a critical factor in road safety, and developing accurate models for detecting it is essential. Transfer learning has been shown to be an effective technique for driver drowsiness detection, as it enables models to leverage large, pre-existing datasets. However, the optimization of hyperparameters in transfer learning models can be challenging, as it involves a large search space. The core purpose of this research is on presenting an approach to hyperparameter tuning in transfer learning for driving fatigue detection based on Bayesian optimization and Random search algorithms. We examine the efficiency of our approach on a publicly available dataset using transfer learning models with the MobileNetV2, Xception, and VGG19 architectures. We explore the impact of hyperparameters such as dropout rate, activation function, the number of units (the number of dense nodes), optimizer, and learning rate on the transfer learning models' overall performance. Our experiments show that our approach improves the performance of the transfer learning models, obtaining cutting-edge results on the dataset for all three architectures. We also compare the efficiency of Bayesian optimization and Random search algorithms in terms of their ability to find optimal hyperparameters and indicate that Bayesian optimization is more efficient in finding optimal hyperparameters than Random search. The results of our study provide insights into the importance of hyperparameter tuning for transfer learning-based driver drowsiness detection using different transfer learning models and can guide the selection of hyperparameters and models for future studies in this field. Our proposed approach can be applied to other transfer learning tasks, making it a valuable contribution to the field of ML.**

***Keywords*—Hyperparameter tuning; driver drowsiness detection; transfer learning; Bayesian optimization; Random search**

I. Introduction

ML models for detecting driver fatigue have shown promise in accurately identifying and alerting drowsy drivers. Transfer learning is a popular approach in developing such models, as it allows leveraging pre-existing datasets to improve model performance. However, optimizing hyperparameters for transfer learning models can be a daunting task given the vast search space involved. In fact, compared to its default value, the model performs far better when the appropriate hyperparameter is used. The hyperparameter changes based on the data collection. The study of hyperparameter optimization for well-known ML models is presented in article [30]. In order to build an effective drowsiness detection system, it is crucial to choose appropriate hyperparameters for the ML model used for this task. Some critical hyperparameters for the drowsiness detection system include dropout rate, activation functions, units (the number of dense nodes), optimizer, and learning rate.

Dropout [12]: In order to prevent overfitting, Dropout is a regularization method that functions by randomly dropping a certain proportion of neurons while training. This reduces overfitting and enhances the model's capacity to make accurate predictions on new data.

Activation function [15]: The activation function is used to provide non-linearity features into the model. Non-linear activation functions are necessary for deep learning (DL) models as they allow the model to learn complex relationships between the input and output data. Common activation functions include Adam, RMSprop, ReLU and tanh. Selecting an activation function will depend on the particular task and the kind of data in use.

Units: The number of dense nodes in a neural network layer is a hyperparameter. Increasing the number of units in a layer increases the model's capacity to learn but also raises the possibility of overfitting. A careful balance must be struck between the number of units and the regularization techniques used to prevent overfitting.

Optimizer [25][34][26]: By utilizing optimization algorithms, researchers can improve the effectiveness of a CNN model for a specific task. The optimizer is used to update the model's weights during training. Common optimizers include SGD, Adam, and RMSprop. When selecting an optimizer, it is important to consider both the task to be performed and the nature of the data.

Learning rate [18]: The learning rate is a crucial factor in determining the size of the weight updates that take place during training. A high learning rate will result in rapid convergence but may also result in the optimizer overshooting the optimal solution. A low learning rate will converge slowly but is less likely to overshoot the optimal solution. The learning rate must be carefully chosen to ensure that the optimizer converges to an optimal solution.

The main contribution of the study is to introduce a method for optimizing hyperparameters in transfer learning for detecting driving fatigue. The approach utilizes both Bayesian optimization and Random search algorithms. The publicly available dataset was used to assess the efficacy of the suggested approach, which was applied to different transfer learning models.

II. Related Works

Driving when fatigued is a serious safety risk, with potentially disastrous consequences. Various methods have been developed for driver drowsiness detection to prevent accidents caused by sleepy or fatigued drivers. One such method is the use of sensors that can detect changes in Physiological-based measures, Physiological-based measures refer to the use of physical signals, to detect changes that indicate drowsiness, such as steering wheel movements [3]. Study [14] presents a low-cost ECG sensor designed for drowsiness detection in drivers. The sensor provides good results in extracting ECG parameters and is used in combination with facial recognition for improved detection in unfavorable conditions. [24] In this study, ECG signals were employed to detect and analyze a driver's condition, with 13 features extracted and trained through SVM, KNN, and Ensemble classifiers. The findings demonstrate high accuracy levels (ranging from 93.1% to 100%) in two-class identification, but a reduced accuracy rate of 58.3% in five-class detection. The study [16] evaluated the efficiency of in-ear EEG in detecting alertness-drowsiness in drivers and compared it with three peripheral signals - ECG, PPG, and GSR. A review article explores various methods that use Electroencephalogram (EEG) signals for detecting and managing driver drowsiness [22]. The paper [23] describes a wearable EEG device that is placed on the driver's forehead and can automatically detect the driver's mental state. [5] This paper proposes a new DL architecture that automatically detects sleepiness from single-channel EEG data using a CNN. Behavioral-based measures refer to the use of observable behaviors and actions to detect drowsiness, such as, The study [29] proposes a method to detect drowsiness in real-time security camera footage by analyzing if someone's eyes are open or closed. This involves identifying the person's face and eyes in the image and applying an extended Sobel operator to detect the shape of the eyelids' curvature. Based on the concavity of the curves, the technique determines whether an individual's eyes are open or closed. The paper proposes a method for detecting driver drowsiness based on eye blink detection. The system detects the eyes using facial landmark extraction and measures the Eye Aspect Ratio to detect blinks. The total number of blinks per minute is compared to a standard value to identify whether the driver is drowsy [17]. A new method [2] for detecting mental fatigue and drowsiness in drivers by using the XSENS motion capture system to analyze head posture movements. A deep neural network using LSTM architecture is utilized to classify driver states based on three dimensions time-series head angular acceleration information, achieving high accuracy with an overall training accuracy of 99.2%. A new system is proposed in [32] for detecting the condition of a driver's eyes and mouth using Retinaface and the residual channel attention network. Results showed high accuracy of 98.962% for eye state classification and 98.561% for mouth state classification on a dataset developed for this study. [4] Describes a drowsiness identification model that integrates face and head pose detection using Dlib models. Recognition of driver fatigue through detecting yawns [20][19][31]. In this work [1][10] a driver drowsiness detection system in real-time that uses facial landmarks and dlib to detect eye movement and yawning. [21] The ratios for eye aspect, mouth opening, and nose length are determined through a process of adaptive thresholding. The study [13] proposes a drowsiness detection method that combines yawning and closing of the eyes using mouth and eye aspect ratios and head pose analysis through optical flow. Two classification techniques, multilayer perception, and K-nearest neighbors were investigated for the prediction of drowsiness. The project [7] detects facial features from a live stream of a driver and then localizes the eyes that are either open or closed. In [31] uses two modules to detect driver fatigue by identifying the condition of the eyes and mouth. The mouth area is identified through depth information while a semi-VGG architecture using CNN networks is employed to detect open or closed eyes. The findings from both detections are integrated and achieved an accuracy of around 90%. The study [11] developed an image-based approach using YOLOv3 CNN and LSTM neural network for detecting driver drowsiness, which showed effectiveness in detecting yawning and blinking time periods with real-time experiments.

III. Transfer Learning Models

Transfer learning is a machine learning (ML) technique to enhance the learning process by utilizing a pre-trained model for a new task. Transfer learning models are neural networks that have been trained on a large dataset, typically for image recognition or natural language processing tasks, and then adapted for a new task with a smaller dataset. The basic idea behind transfer learning is that a model that has learned to recognize certain features in one domain can be re-purposed to recognize similar features in another domain. There are several approaches to implementing transfer learning models, but they generally involve modifying the pretrained model to suit the new task. This might involve adding new layers to the model, changing the total of neurons in the existing layers, or fine-tuning the weights of the existing layers. One of the most evident advantages of transfer learning is that it can considerably minimize the volume of data needed to train a fresh model since many of the relevant characteristics have already been learned by the pre-existing model. This can be particularly useful for tasks where there is limited labeled data available. Transfer learning can be advantageous in that it helps to enhance the performance of a model since the pretrained model has already learned to recognize a variety of crucial features in the input data. Examples of transfer learning in various fields include Computer Vision: Pretrained models can be fine-tuned on a smaller dataset of medical images to detect specific types of tumors in X-ray images. Natural Language Processing: Pretrained models can be further adjusted on a smaller dataset of news articles to detect fake news. Speech Recognition: Pretrained models for speech recognition can be fine-tuned on a smaller dataset of spoken commands to control a smart home device.

The study aims to use is to use Bayesian Optimization and Random Search to find hyperparameters for the Transfer Learning model, including VGG19 [28], MobileNetV2 [27] and Xception [8]. The experiments carried out for this research are listed in Table I and the search space for hyperparameters is given in Table II.

TABLE I. THE TECHNIQUE INVOLVES LEVERAGING A PRE-TRAINED NETWORK FOR FEATURE EXTRACTION AND THEN TUNING THE MODEL THROUGH FINE-TUNING

Models	**VGG19**	**Xception**	**MobileNetV2**
Usage of pre-trained	The technique involves leveraging a pre-trained network for feature extraction and then tuning the model through fine-tuning to improve its effectiveness		
Description	Remove the classification layer at the top of the pre-trained model. A new model head is constructed on top of the base model. This includes an AveragePooling2D layer. A Flatten layer to transform the output into a vector. A layer that is Dense and specifies the number of units and activation function. A Dropout layer to prevent overfitting. A Dense layer which is typically used in the final output layer of a neural network with a softmax activation function is added to output the probability distribution over four classes. During the training process, the weights of the base model layers are frozen to prevent updating, while only the weights of the new head model are updated		
Hyperparameter	Dropout rate, Activation function, Number of units, Optimizer, and Learning rate		

TABLE II. THE SEARCH SPACE FOR HYPERPARAMETERS

Hyper parameter	**Search space**
Dropout rate	values=[0.0, 0.1, 0.2, 0.3, 0.4, 0.5]
Activation function	values=['softplus', 'softmax', 'softsign', 'relu', 'hard_sigmoid', 'tanh', 'sigmoid']
Number of units	values=[32, 64, 128, 256, 512, 1024, 2048]
Optimizer	values=['Adadelta', 'Adagrad', 'Adam', 'RMSprop', 'SGD', 'Adamax', 'Nadam']
Learning rate	values=[1e-1, 1e-2, 1e-3, 1e-4, 1e-5]

IV. HYPERPARAMETER OPTIMIZATION

The main goal of hyperparameter optimization is to automate the process of hyperparameter tuning and make it more efficient. The overall objective of a hyper-parameter optimization problem is to achieve [30].

$$x^* = \arg \underset{x \in X}{min} f(x) \tag{1}$$

Where $f(x)$ is the objective function; x^* is the configuration of hyperparameters that generates the optimal value for $f(x)$; the hyperparameter x can assume any element in the search space X.

A. Grid Search

Grid search [30] is a popular method used for exploring the configuration space of hyperparameters. This method involves evaluating all possible combinations of hyperparameters. Grid search is a straightforward approach that can be easily implemented and parallelized. However, it becomes less efficient when the number of hyperparameters increases, as the number of evaluations required to search the entire configuration space grows exponentially.

B. Random Search

Random search [30] [6] [33] is a simple and straightforward hyperparameter optimization method that is commonly used in ML. The approach involves randomly sampling hyperparameters from a specified search space, and evaluating the resulting models on a validation set to determine the performance of each set of hyperparameters.

One of the main advantages of Random search is its simplicity and ease of implementation. It requires minimal tuning and can be easily parallelized, allowing for efficient use of computational resources. Additionally, Random search can often outperform more sophisticated optimization methods for low-dimensional hyperparameter spaces. A significant drawback of both Grid search and Random search is that each evaluation made throughout the search process is independent of any prior evaluations. Therefore, while selecting the next set of hyperparameters to assess, these approaches do not consider the outcomes of earlier evaluations.

C. Bayesian Optimization

Bayesian optimization [30][9] is a probabilistic approach to global optimization that is commonly used in ML, optimization, and experimental design. The goal of Bayesian optimization is to find the optimal set of hyperparameters that maximizes a given objective function, such as the accuracy of DL and ML models or the efficiency of a manufacturing process.

Several models, such as the Gaussian Process (GP), Random Forest (RF), and Tree-structured Parzen Estimators (TPE) models, can be used as the surrogate function in Bayesian optimization to simulate the distribution of the objective function, to approximate the unknown objective function. Iteratively updating the surrogate model with new observations of the objective function as the optimization process progresses, allowing for more accurate predictions of the optimal set of hyperparameters.

One of the main advantages of Bayesian optimization is its ability to efficiently optimize complex, high-dimensional, and non-convex functions with noisy and expensive evaluations. This makes it particularly useful for optimizing the hyperparameters of DL and ML models, which often involve complex and high-dimensional spaces. Furthermore, Bayesian optimization is able to handle constraints and multi-objective optimization problems.

V. METHODOLOGY

A. Dataset and Data Preparation

The dataset used in this study consists of a total of 3064 images, which were collected for the purpose of driver drowsiness detection. The dataset is composed of four categories: eyes closed, eyes open, yawning, and no yawning. There are 726 images of eyes closed, which were obtained by capturing the eye region of subjects when their eyes were closed. Similarly, there are 726 images of Eye Open, which were obtained when subjects had their eyes open. The third category, Yawn, consists of 820 images, which were captured when subjects were yawning, and the fourth category, No Yawn, consists of 812 images, which were captured when subjects were not yawning. The dataset was partitioned into three distinct sets: training set, testing set, and validation set - in a random manner. The distribution of the dataset can be found in Fig. 1 and Fig. 2.

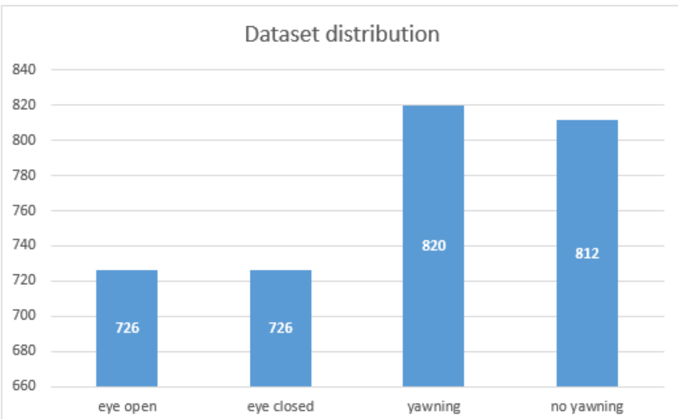


Fig. 1. A dataset distribution.

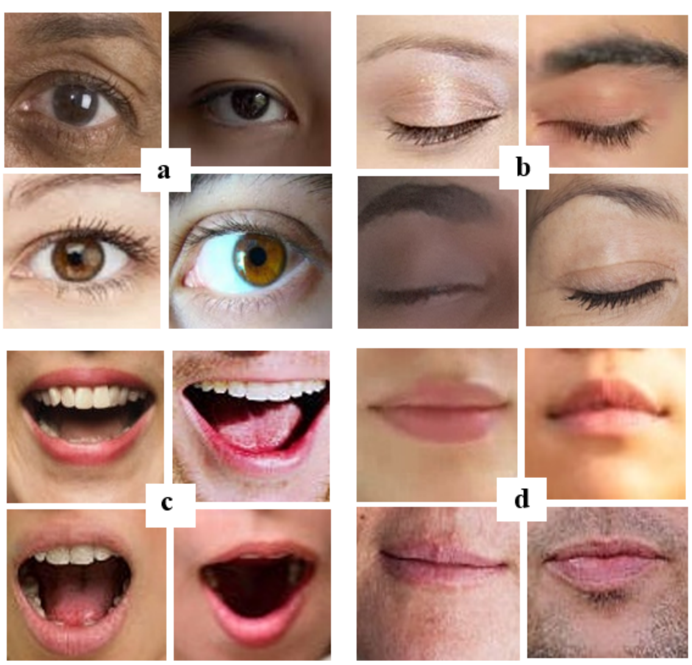


Fig. 2. Samples of drowsiness disease images. (a) Eye open, (b) Eye closed, (c) Yawn (d) No yawn.

B. Model Evaluation Metrics

In this research, the effectiveness of the DL models was examined using a variety of metrics. These metrics include Precision, Recall, F1-score, and Accuracy. Accuracy was used to measure the overall effectiveness of the models in predicting the target variable. Precision was used to measure the proportion of true positive results among all the positive predictions is being calculated, while recall was used to measure the ratio of correctly predicted positive instances out of all actual positive instances in the dataset is being evaluated. F1-score, which combines both precision and recall, was used to provide a balanced view of the model's performance, especially in cases where the classes are imbalanced. By utilizing multiple evaluation metrics, were able to achieve an in-depth comprehension of the model's performance.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{2}$$

$$Precision = \frac{TP}{TP + FP} \tag{3}$$

$$Recall = \frac{TP}{TP + FN} \tag{4}$$

$$F_1 - Score = \frac{Precision * Recall}{Precision + Recall} \tag{5}$$

In which, TP: True Positive, TN: True Negative, FP: False Positive, FN: False Negative.

C. Results

In order to train the DL model, the images were altered in size to either 224 x 224 for the VGG19, MobileNetV2, and Xception architectures. Removing the original fully connected Layer in MobileNetV2, VGG199, and Xception base network and changing them with new fully-connected Layers, then fine-tuning to a dataset for detecting drowsiness. The models of architecture utilized in this research are displayed in Fig. 3. During our experiments, we employed the Bayesian optimization and Random search techniques to adjust a set of five parameters that are fine-tuned within the neural networks. Specifically, we fine-tuned the dropout rate, activation function, units (number of dense nodes), optimizer, and learning rate.

Three models, VGG19, Xception, and MobileNetV2, were trained and tested on a given dataset. Bayesian optimization and Random search were employed to fine-tune the models. VGG19 achieved an accuracy of 0.97039 with Bayesian optimization, and 0.96875 with Random search. Xception obtained an accuracy of 0.98355 with both Bayesian optimization and Random search, while MobileNetV2 achieved the highest accuracy of 0.98684 with Bayesian optimization and 0.98190 with Random search. Overall, the results show that MobileNetV2 outperformed the other models with the highest accuracy in Bayesian optimization. The optimized hyperparameters found using Bayesian Optimization and Random Search for VGG19, Xception, and MobileNetV2 models are shown in Table III and is used to conduct various test scenarios described in Table I.

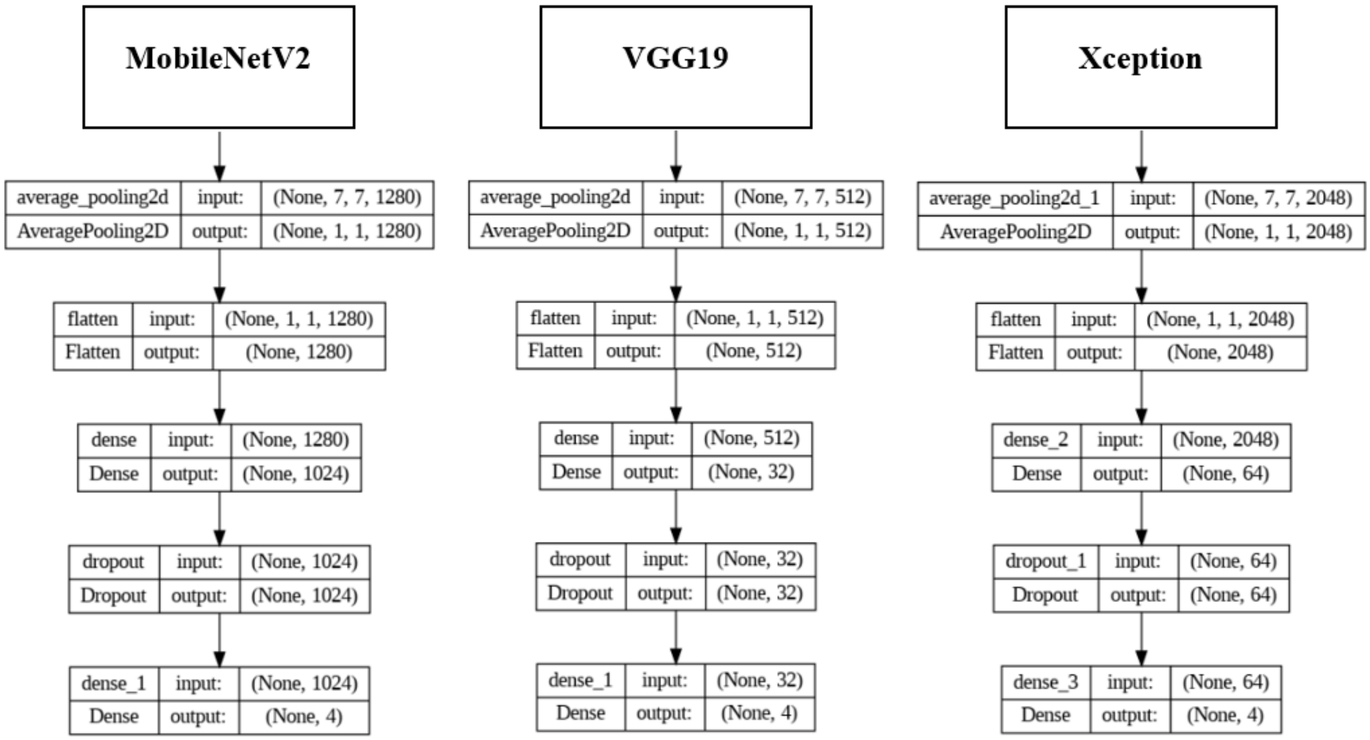


Fig. 3. Removing the original fully connected Layer in MobileNetV2, VGG199, and Xception base network and changing them with new fully connected Layers.

1) VGG19: Fig. 4 displays the optimized confusion matrices for the VGG19 model that have been optimized using Bayesian optimization and Random search techniques. The VGG19 model's training and validation performance metrics including loss and accuracy are shown in Fig. 5. To get these results, Bayesian optimization and Random search strategies were also used. The precision, recall, F1-score, and accuracy were achieved during the evaluation of this model on each class using the best hyperparameters determined by Bayesian optimization and Random search presented in Table IV.

2) Xception: The optimized confusion matrices for the Xception model are shown in Fig. 6, which were produced using both Bayesian optimization and Random search methods. Additionally, using the above optimization techniques, Fig. 7 shows the Xception model's performance during training and validation. The precision, recall, F1-score, and accuracy achieved during the evaluation of this model on each class using the best hyperparameters determined by Bayesian optimization and Random search are presented in Table V.

3) MobileNetV2: Both Bayesian optimization and Random search methods were utilized to create the optimized confusion matrices for the MobileNetV2 model, as presented in Fig. 8. Additionally, Fig. 9 showcases the performance during training and validation of the MobileNetV2 model, obtained through the application of the aforementioned optimization methods. The precision, recall, F1-score, and accuracy were achieved during the evaluation of this model on each class using the best hyperparameters determined by Bayesian optimization and Random search presented in Table VI.

The classification performance of the three optimized models is displayed in Fig. 10. In terms of precision, the MobileNetV2 model performs better than all the others. Compared to Random search, Bayesian optimization is more effective at obtaining optimal hyperparameters.

VI. Conclusion

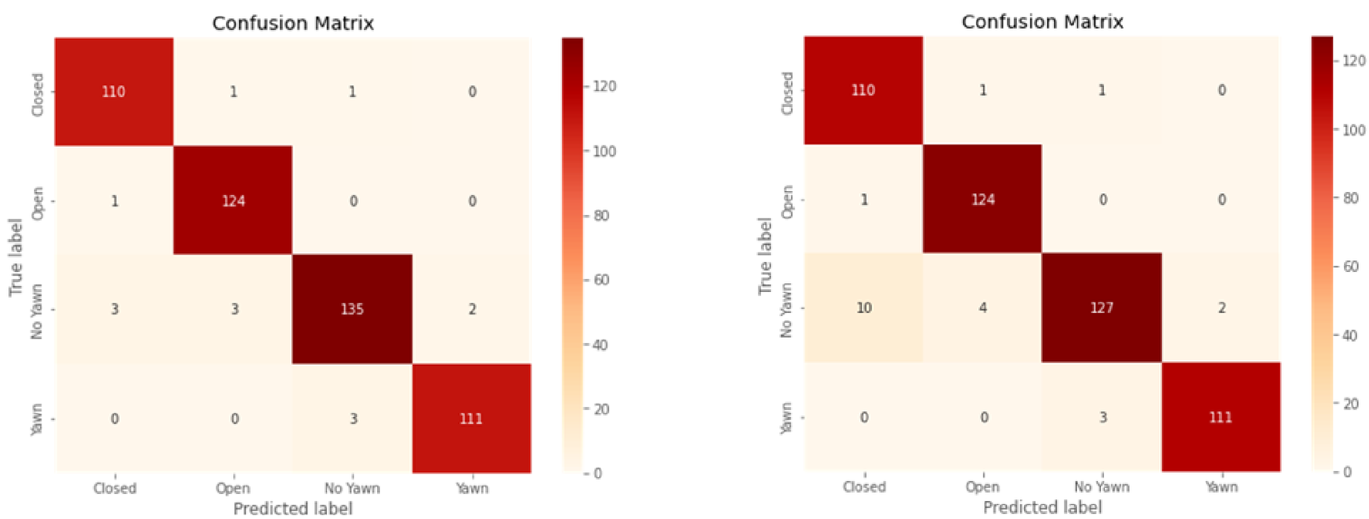
In this study, we proposed an approach for hyperparameter tuning in transfer learning for driver drowsiness detection using Bayesian optimization and Random search algorithms. We evaluated our approach on a publicly available dataset using transfer learning models with the MobileNetV2, Xception, and VGG19 architectures. We explored the impact of various hyperparameters including dropout rate, activation function, number of units, optimizer, and learning rate on the performance of the transfer learning models.

Our experiments show that our approach improves the effectiveness of the transfer learning models, obtaining cutting-edge results on the dataset for all three architectures. We also compared the performances of Bayesian optimization and Random search algorithms in terms of their ability to find optimal hyperparameters and demonstrated that Bayesian optimization is more efficient in finding optimal hyperparameters than Random search.

Our findings highlight the importance of hyperparameter tuning in transfer learning-based driver drowsiness detection and provide insights into the impact of different hyperparameters on the effectiveness of the transfer learning models. Furthermore, our proposed approach can be applied to other transfer learning tasks, making it a valuable contribution to the field of ML.

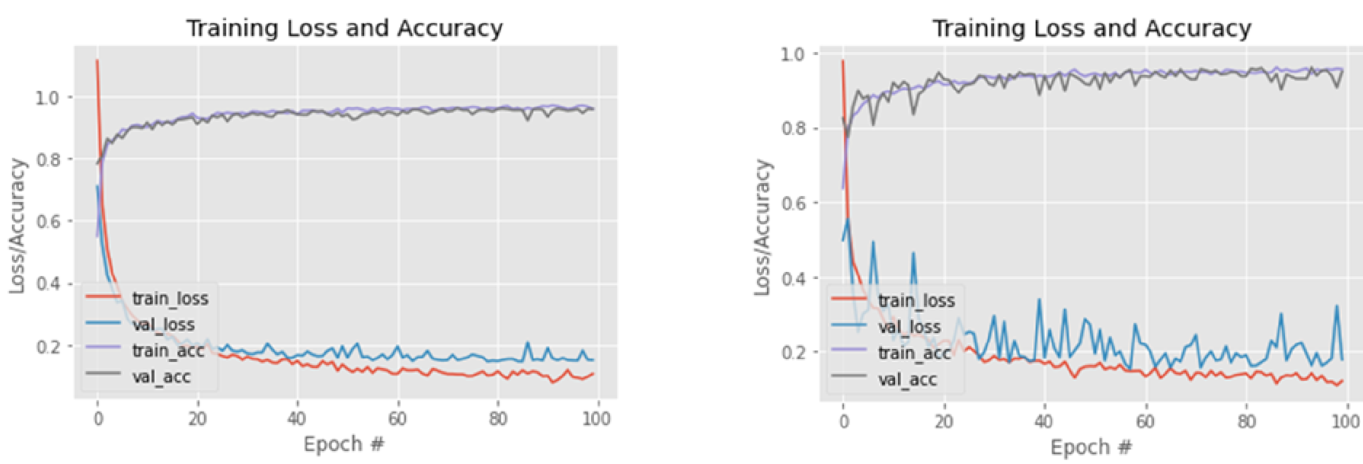
TABLE III. HYPERPARAMETER VALUES FOUND USING BAYESIAN OPTIMIZATION AND RANDOM SEARCH

Models	Dropout rate	Activation function	Number of units	Optimizer	Learning rate	Accuracy
VGG19						
Bayesian Optimization	0.2	Elu	32	Adam	0.001	0.97039
Random Search	0.1	Tanh	1024	RMSprop	0.001	0.96875
Xception						
Bayesian Optimization	0.1	Elu	64	RMSprop	0.01	0.98355
Random Search	0.5	Tanh	128	Adam	0.001	0.98355
MobileNetV2						
Bayesian Optimization	0.1	Tanh	1024	RMSprop	0.001	0.98684
Random Search	0.5	Elu	1024	Adagrad	0.1	0.98190



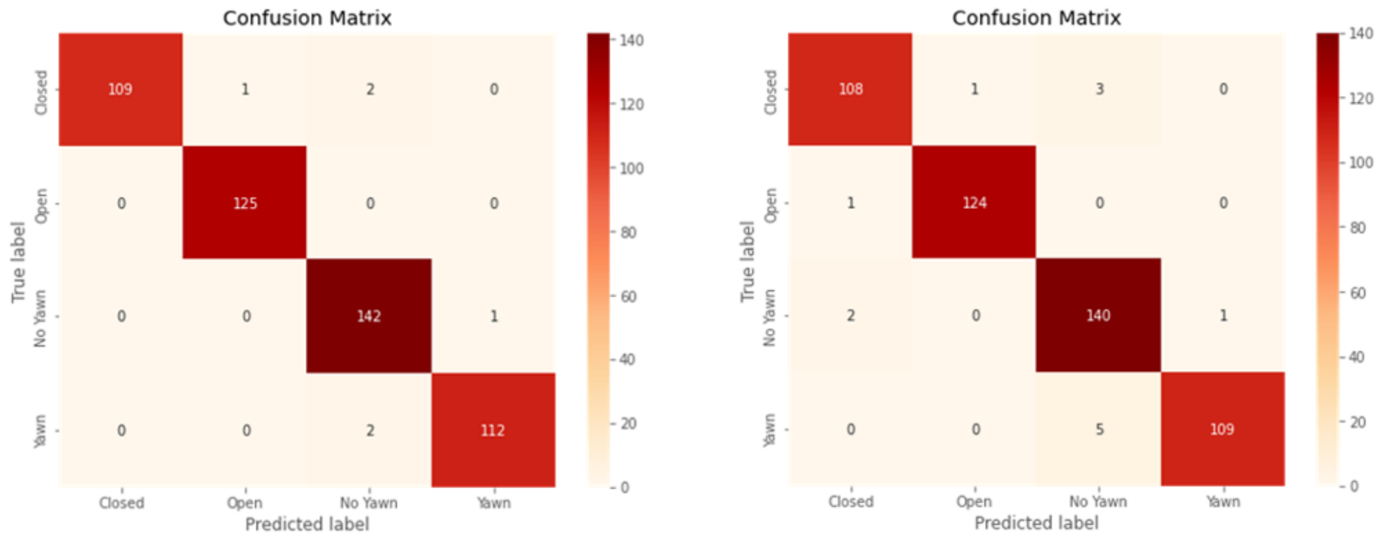
VGG19 (a) VGG19 (b)

Fig. 4. Confusion matrix of VGG19. VGG19 (a) The ideal hyperparameter values are chosen through the utilization of Bayesian optimization. VGG19 (b) The ideal hyperparameter values are chosen through the utilization of Random search.



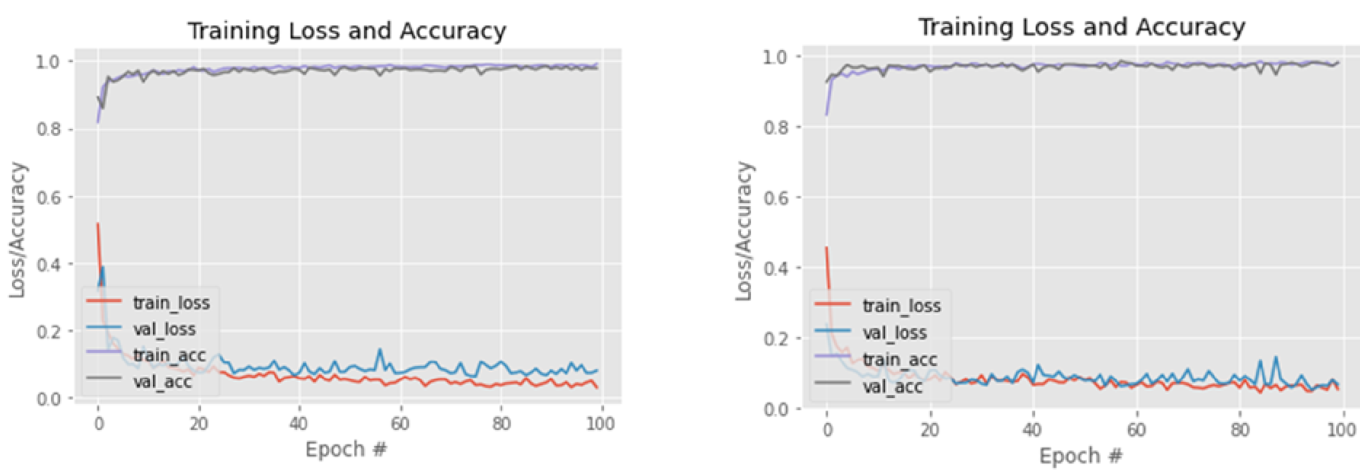
VGG19 (a) VGG19 (b)

Fig. 5. Loss and accuracy plots of VGG19. VGG19 (a) The ideal hyperparameter values are chosen through the utilization of Bayesian optimization. VGG19 (b) The ideal hyperparameter values are chosen through the utilization of Random search.



Xception (a) Xception (b)

Fig. 6. Confusion matrix of Xception. Xception (a) The ideal hyperparameter values are chosen through the utilization of Bayesian optimization. Xception (b) The ideal hyperparameter values are chosen through the utilization of Random search.



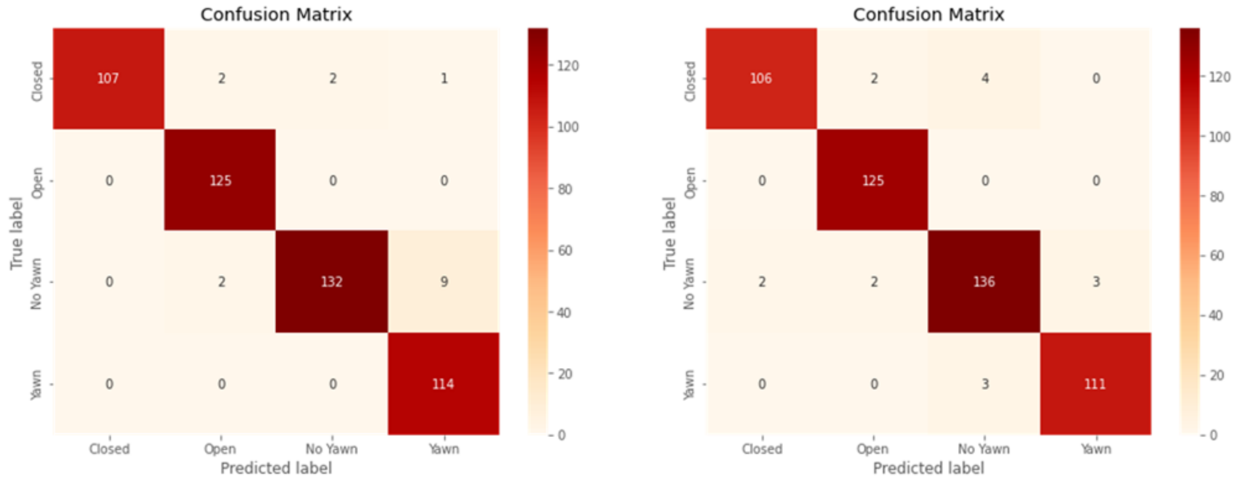
Xception (a) Xception (b)

Fig. 7. Loss and accuracy plots of Xception. Xception (a) The ideal hyperparameter values are chosen through the utilization of Bayesian optimization. Xception (b) The ideal hyperparameter values are chosen through the utilization of Random search.

Overall, our study emphasizes the significance of selecting appropriate hyperparameters to achieve optimal performance in transfer learning models for detecting driver sleepiness and provides a framework for doing so. The results of this study can guide the selection of hyperparameters and models for future research in this area.

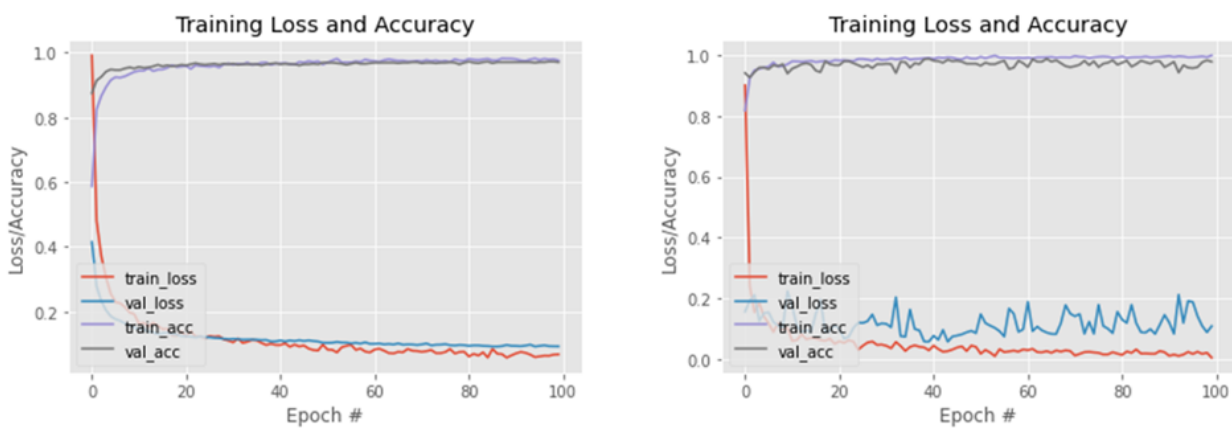
FUTURE WORKS

Further investigations can be carried out to explore the impact of other hyperparameters, such as weight decay and batch size, on the effectiveness of transfer learning models for detecting driver sleepiness. Furthermore, the combination of multiple optimization techniques, such as Bayesian optimization, Random search, and Hyperband can be researched to further improve the efficiency of hyperparameter tuning. Finally, the proposed approach can be extended to real-time sleepiness detection for drivers using mobile devices. The implementation of optimized transfer learning models on such devices can enable real-time monitoring of driver drowsiness, thus enhancing road safety.



MobileNetV2 (a) MobileNetV2 (b)

Fig. 8. Confusion matrix of MobileNetV2. MobileNetV2 (a) The ideal hyperparameter values are chosen through the utilization of Bayesian optimization. MobileNetV2 (b) The ideal hyperparameter values are chosen through the utilization of Random search.



MobileNetV2 (a) MobileNetV2 (b)

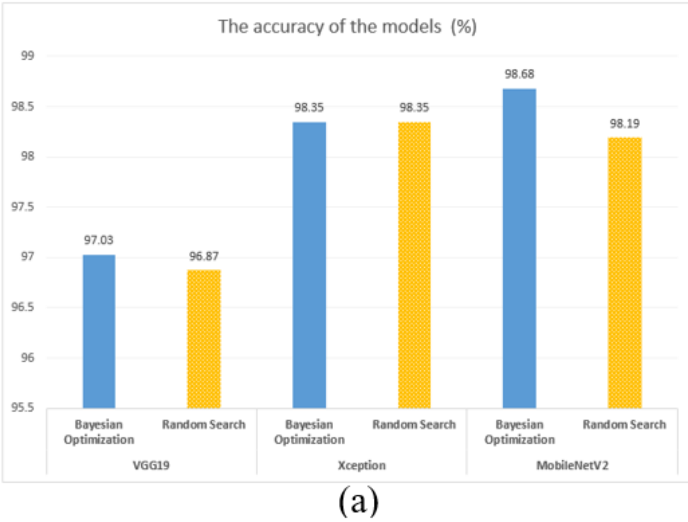
Fig. 9. Loss and accuracy plots of MobileNetV2. MobileNetV2 (a) The ideal hyperparameter values are chosen through the utilization of Bayesian optimization. MobileNetV2 (b) The ideal hyperparameter values are chosen through the utilization of Random search.

TABLE IV. THE PRECISION, RECALL, F1-SCORE, AND ACCURACY ACHIEVED DURING THE EVALUATION OF THE VGG-19 MODEL USING THE BEST HYPERPARAMETERS DETERMINED BY BAYESIAN OPTIMIZATION AND RANDOM SEARCH

VGG19	Precision	Recall	F1-score	Accuracy
Bayesian Optimization				
Closed	0.9649	0.9821	0.9734	0.9821
Open	0.9687	0.9920	0.9802	0.9920
No Yawn	0.9712	0.9440	0.9574	0.9440
Yawn	0.9823	0.9736	0.9779	0.9736
Random Search				
Closed	0.9090	0.9821	0.9442	0.9821
Open	0.9612	0.9920	0.9763	0.9920
No Yawn	0.9694	0.8881	0.9270	0.8881
Yawn	0.9823	0.9736	0.9779	0.9736

TABLE V. THE PRECISION, RECALL, F1-SCORE, AND ACCURACY ACHIEVED DURING THE EVALUATION OF THE XCEPTION MODEL USING THE BEST HYPERPARAMETERS DETERMINED BY BAYESIAN OPTIMIZATION AND RANDOM SEARCH

Xception	Precision	Recall	F1-score	Accuracy
Bayesian Optimization				
Closed	1.0000	0.9732	0.9864	0.9732
Open	0.9920	1.0000	0.9960	1.0000
No Yawn	0.9726	0.9930	0.9826	0.9930
Yawn	0.9911	0.9824	0.9867	0.9824
Random Search				
Closed	0.9729	0.9642	0.9686	0.9642
Open	0.9920	0.9920	0.9920	0.9920
No Yawn	0.9459	0.9790	0.9621	0.9790
Yawn	0.9909	0.9561	0.9732	0.9561



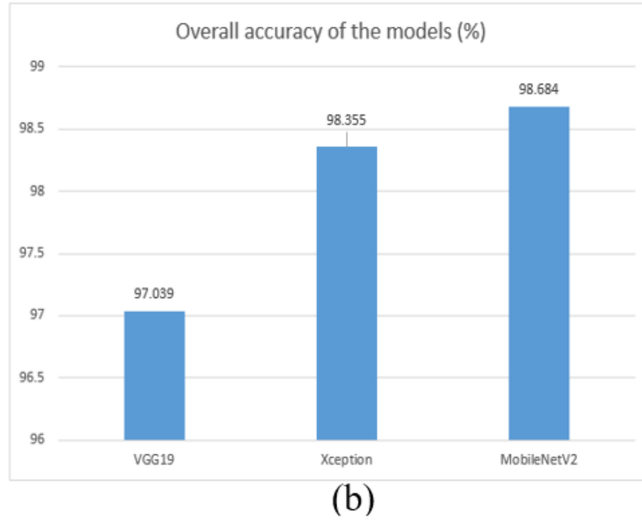


Fig. 10. (a) The accuracy of the models based on hyperparameters found by two methods: Bayesian Optimization and Random Search; (b) Overall accuracy of the models.

TABLE VI. The precision, recall, F1-score, and accuracy achieved during the evaluation of the MobileNetV2 model using the best hyperparameters determined by Bayesian optimization and Random search

MobileNetV2	Precision	Recall	F1-score	Accuracy
Bayesian Optimization				
Closed	1.0000	0.9553	0.9771	0.9553
Open	0.9689	1.0000	0.9842	1.0000
No Yawn	0.9850	0.9230	0.9530	0.9230
Yawn	0.9193	1.0000	0.9579	1.0000
Random Search				
Closed	0.9814	0.9464	0.9636	0.9464
Open	0.9689	1.0000	0.9842	1.0000
No Yawn	0.9510	0.9510	0.9510	0.9510
Yawn	0.9736	0.9736	0.9736	0.9736

References

[1] Ali Mansour Al-Madani, Ashok T Gaikwad, Vivek Mahale, Zeyad AT Ahmed, and Ahmed Abdullah A Shareef. Real-time driver drowsiness detection based on eye movement and yawning using facial landmark. In *2021 International Conference on Computer Communication and Informatics (ICCCI)*, pages 1–4. IEEE, 2021.

[2] Shahzeb Ansari, Fazel Naghdy, Haiping Du, and Yasmeen Naz Pahnwar. Driver mental fatigue detection based on head posture using new modified relu-bilstm deep neural network. *IEEE Transactions on Intelligent Transportation Systems*, 23(8):10957–10969, 2021.

[3] Sadegh Arefnezhad, Sajjad Samiee, Arno Eichberger, and Ali Nahvi. Driver drowsiness detection based on steering wheel data applying adaptive neuro-fuzzy feature selection. *Sensors*, 19:943, 02 2019.

[4] Athira Babu, Shruti Nair, and K Sreekumar. Driver's drowsiness detection system using dlib hog. In *Ubiquitous Intelligent Systems: Proceedings of ICUIS 2021*, pages 219–229. Springer, 2022.

[5] Venkata Phanikrishna Balam, Venkata Udaya Sameer, and Suchismitha Chinara. Automated classification system for drowsiness detection using convolutional neural network and electroencephalogram. *IET Intelligent Transport Systems*, 15(4):514–524, 2021.

[6] James Bergstra and Yoshua Bengio. Random search for hyper-parameter optimization. *Journal of machine learning research*, 13(2), 2012.

[7] Varun Chaudhary, Ziyad Dalwai, and Vikram Kulkarni. Intelligent distraction and drowsiness detection system for automobiles. In *2021 International Conference on Intelligent Technologies (CONIT)*, pages 1–4. IEEE, 2021.

[8] François Chollet. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1251–1258, 2017.

[9] Nando de Freitas. A tutorial on bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. 2009.

[10] Wanghua Deng and Ruoxue Wu. Real-time driver-drowsiness detection system using facial features. *Ieee Access*, 7:118727–118738, 2019.

[11] Farnoosh Faraji, Faraz Lotfi, Javad Khorramdel, Ali Najafi, and Ali Ghaffari. Drowsiness detection based on driver temporal behavior using a new developed dataset. *arXiv preprint arXiv:2104.00125*, 2021.

[12] Christian Garbin, Xingquan Zhu, and Oge Marques. Dropout vs. batch normalization: an empirical study of their impact to deep learning. *Multimedia Tools and Applications*, 79:12777–12815, 2020.

[13] Aicha Ghourabi, Haythem Ghazouani, and Walid Barhoumi. Driver drowsiness detection based on joint monitoring of yawning, blinking and nodding. In *2020 IEEE 16th International Conference on Intelligent Computer Communication and Processing (ICCP)*, pages 407–414. IEEE, 2020.

[14] Markus Gromer, David Salb, Thomas Walzer, Natividad Martínez Madrid, and Ralf Seepold. Ecg sensor for detection of driver's drowsiness. *Procedia Computer Science*, 159:1938–1946, 2019.

[15] Soufiane Hayou, Arnaud Doucet, and Judith Rousseau. On the impact of the activation function on deep neural networks training. In *International conference on machine learning*, pages 2672–2680. PMLR, 2019.

[16] Taeho Hwang, Miyoung Kim, Seunghyeok Hong, and Kwang Suk Park. Driver drowsiness detection using the in-ear eeg. In *2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 4646–4649. IEEE, 2016.

[17] Arafat Islam, Naimur Rahaman, and Md Atiqur Rahman Ahad. A study on tiredness assessment by using eye blink detection. *Jurnal Kejuruteraan*, 31(2):209–214, 2019.

[18] Jennifer Jepkoech, David Muchangi Mugo, Benson K Kenduiywo, and Edna Chebet Too. The effect of adaptive learning rate on the accuracy of neural networks. 2021.

[19] Zhuoni Jie, Marwa Mahmoud, Quentin Stafford-Fraser, Peter Robinson, Eduardo Dias, and Lee Skrypchuk. Analysis of yawning behaviour in spontaneous expressions of drowsy drivers. In *2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018)*, pages 571–576. IEEE, 2018.

[20] Mateusz Knapik and Bogusław Cyganek. Driver's fatigue recognition based on yawn detection in thermal images. *Neurocomputing*, 338:274–292, 2019.

[21] Ashish Kumar and Rusha Patra. Driver drowsiness monitoring system using visual behaviour and machine learning. In *2018 IEEE Symposium on Computer Applications & Industrial Electronics (ISCAIE)*, pages 339–344. IEEE, 2018.

[22] Gang Li and Wan-Young Chung. Electroencephalogram-based approaches for driver drowsiness detection and management: a review. *Sensors*, 22(3):1100, 2022.

[23] Jennalyn N Mindoro, Cherry D Casuat, Alvin Sarraga Alon, Mon Arjay F Malbog, and Julie Ann B Susa. Drowsy or not? early drowsiness detection utilizing arduino based on electroencephalogram (eeg) neuro-signal. *International Journal*, 9(2), 2020.

[24] Suganiya Murugan, Jerritta Selvaraj, and Arun Sahayadhas. Detection and analysis: driver state with electrocardiogram (ecg). *Physical and engineering sciences in medicine*, 43(2):525–537, 2020.

[25] Luyl-Da Quach, Nguyen Quoc Khang, Anh Nguyen Quynh, and Tran Ngoc Hoang. Evaluation of the efficiency of the optimization algorithms for transfer learning on the rice leaf disease dataset. *International Journal of Advanced Computer Science and Applications*, 13(10), 2022.

[26] Luyl-Da Quach, Anh Nguyen Quynh, Khang Nguyen Quoc, and Nghe Nguyen Thai. Using optimization algorithm to improve the accuracy of the cnn model on the rice leaf disease dataset. In *Information Systems for Intelligent Systems: Proceedings of ISBM 2022*, pages 535–544. Springer, 2023.

[27] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018.

[28] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[29] Muhammad Tayab Khan, Hafeez Anwar, Farman Ullah, Ata Ur Rehman, Rehmat Ullah, Asif Iqbal, Bok-Hee Lee, and Kyung Sup Kwak. Smart real-time video surveillance platform for drowsiness detection based on eyelid closure. *Wireless communications and mobile computing*, 2019, 2019.

[30] Li Yang and Abdallah Shami. On hyperparameter optimization of machine learning algorithms: Theory and practice. *Neurocomputing*, 415:295–316, 2020.

[31] Mina Zohoorian Jafari Yazdi and Mohsen Soryani. Driver drowsiness detection by yawn identification based on depth information and active contour model. In *2019 2nd International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT)*, volume 1, pages 1522–1526. IEEE, 2019.

[32] Mu Ye, Weiwei Zhang, Pengcheng Cao, and Kangan Liu. Driver fatigue detection based on residual channel attention network and head pose estimation. *Applied Sciences*, 11(19):9195, 2021.

[33] Zelda B Zabinsky et al. Random search algorithms. *Department of Industrial and Systems Engineering, University of Washington, USA*, 2009.

[34] Raniah Zaheer and Humera Shaziya. A study of the optimization algorithms in deep learning. In *2019 third international conference on inventive systems and control (ICISC)*, pages 536–539. IEEE, 2019.