electronics

MDPI

Article

MobileNet-Based Architecture for Distracted Human Driver Detection of Autonomous Cars

Mahmoud Abdelkader Bashery Abbass [1,2] and Yuseok Ban [1,*]

[1] School of Electronics Engineering, Chungbuk National University, Cheongju-si 28644, Republic of Korea; mahmoud.gohar1992@chungbuk.ac.kr or mahmoud.gohar1992@m-eng.helwan.edu.eg
[2] Mechanical Power Department, Faculty of Engineering—Mataria, Helwan University, Cairo 11772, Egypt
* Correspondence: ban@cbnu.ac.kr; Tel.: +82-43-261-2475

Abstract: Distracted human driver detection is an important feature that should be included in most levels of autonomous cars, because most of these are still under development. Hereby, this paper proposes an architecture to perform this task in a fast and accurate way, with a full declaration of its details. The proposed architecture is mainly based on the MobileNet transfer learning model as a backbone feature extractor, then the extracted features are averaged by using a global average pooling layer, and then the outputs are fed into a combination of fully connected layers to identify the driver case. Also, the stochastic gradient descent (SGD) is selected as an optimizer, and the categorical cross-entropy is the loss function through the training process. This architecture is performed on the State-Farm dataset after performing data augmentation by using shifting, rotation, and zooming. The architecture can achieve a validation accuracy of 89.63%, a validation recall of 88.8%, a validation precision of 90.7%, a validation f1-score of 89.8%, a validation loss of 0.3652, and a prediction time of about 0.01 seconds per image. The conclusion demonstrates the efficiency of the proposed architecture with respect to most of the related work.

Keywords: Distracted Human Driver Detection; Autonomous Cars; MobileNet Model; State-Farm Dataset

Citation: Abbass, M.A.B.; Ban, Y. MobileNet-Based Architecture for Distracted Human Driver Detection of Autonomous Cars. *Electronics* **2024**, *13*, 365. https://doi.org/10.3390/electronics13020365

Academic Editor: Felipe Jiménez

Received: 17 November 2023
Revised: 25 December 2023
Accepted: 12 January 2024
Published: 15 January 2024

1. Introduction

Traffic accidents frequently lead to injuries, the impairment of physical capabilities, and even fatalities. Every day, hundreds of humans lose their lives in traffic accidents, which have become increasingly prevalent due to driving distractions such as using a cell phone and interacting with passengers. The World Health Organization estimates that traffic accidents result in up to 50 million injuries and 1.35 million deaths every year, with an average of 64 deaths every day [1]. Furthermore, distractions while driving account for over 80% of all car accidents [2]. With this, global technological efforts are being directed toward devising viable solutions to meet safety requirements. Thus, the task of driving is increasingly being redirected towards autonomous driving systems to decrease reliance on human drivers. However, the existing technology for the autonomous system is not yet sufficiently advanced to operate without human interaction, so human attention to the road and the behavior of the autonomous system is still required. Therefore, the capacities of a human driver to monitor and remain alert continue to be essential during the operation of the autonomous driving system [3–6].

Autonomous vehicles are those that can drive themselves without the need for human involvement. In self-driving automobiles, the level of autonomy refers to how much the car can accomplish on its own and how much the human driver must be involved. There are many societies (e.g., The Society of Automotive Engineers (SAE), the International Organization of Motor Vehicle Manufacturers (OICA), the German Federal Highway Research Institute (BASt), and the US National Highway Traffic Safety Administration (NHTSA))

that classify driving automation into six categories (as shown in Table 1), ranging from 0 (completely manual) to 5 (completely autonomous) [7–9].

Table 1. Descriptions of the levels of autonomous driving systems.

Level	Name	Autonomous System	Human Driver
0	No driving automation	The car as a whole does not have any autonomous functions	The human driver is in full control of all elements of driving
1	Driver Assistance	The car features a single automated system for driver assistance, including steering or accelerating	The human driver must continue to monitor the road and take control of the car at all times
2	Partial driving automation	The car has two or more automated driver assistance systems, such as steering and accelerating. Under some situations, the car can manage both steering and speed	The human driver must remain alert and ready to take action
3	Conditional driving automation	The car can monitor its surroundings and make smart judgments by itself, such as passing another vehicle. Under some conditions, the car can drive itself	The human driver must remain attentive and take control if the system fails or faces a scenario it cannot manage
4	High driving automation	Under normal situations, the car can drive itself without human assistance. The car can perform all driving activities and scenarios within its operational design domain, which is a set region or scenario in which the car may drive safely and lawfully	The human driver is not required to pay attention or take over, but can still drive manually if wanted
5	Full Driving Automation	A car can drive itself in any situation, with no boundaries or constraints. The car can handle every driving duty or scenario, wherever and at any time, without human intervention or supervision	The human driver is entirely optional and can just sit in the passenger seat

Based on the various levels of autonomous driving systems, human driver attention is required from level zero to level three. So, distracted driver detection is required to increase the safety of the autonomous driving task, especially from level zero to level three.

The distracted driver detection task has a lot of challenges. Firstly, the task needs an accurate and robust algorithm to detect the various behaviors of a driver and identify each one accurately. Secondly, due to the high speeds of cars, the algorithm must be fast in the detection process to reduce the response time for the overall system [10,11]. For these challenges, the MobileNet model is selected to perform the task through this paper, because this model has a small number of parameters compared to other transfer learning models (e.g., VGG and ResNet). In other words, the time needed to generate a prediction using the MobileNet model is significantly shorter compared to the extensive computation required by the other transfer learning architectures, as mentioned in previous work [12–14]. In addition, the proposed work is performed on the State-Farm dataset (as shown in Table 2) [15], which contains ten distraction classes for various drivers, to obtain a robust, fast, and accurate model.

1.1. Related Work

In this section, the related work for the detection of a distracted human driver is broadly reviewed, and previous works that have already studied for distracted driver detection using the MobileNet architecture and the State-Farm data are employed. In addition, the details and the results of each paper are demonstrated in Table 3 and compared with the result of our proposed architecture that outperforms those of most previous works.

Generally, there are three main techniques to handle real-life problems such as the distracted human driver detection: (1) RGB-based techniques: the methods that use the color information from a camera to analyze the driver's behavior [16,17]; (2) RGB + depth-based technique: the methods that use both the color and the depth information from a camera to analyze the driver's behavior [18]; (3) RGB + Depth + InfraRed-based technique:

the methods that use the color, depth, and infrared information from a camera to analyze the driver's behavior [19–21].

Table 2. State-Farm distracted driver detection dataset specifications.

Class	Label	Image Number
C0	Safe driving	2489
C1	Texting with right hand	2267
C2	Talking on the phone with right hand	2317
C3	Texting with left hand	2346
C4	Talking on the phone with left hand	2326
C5	Operating the radio	2312
C6	Drinking	2325
C7	Reaching behind	2002
C8	Hair and makeup	1911
C9	Talking to passenger	2129

The presented paper focused on the RGB-based technique, due to four reasons: (1) Universality and availability: RGB cameras are standard in most devices, making this method more accessible and cost-effective for widespread implementation. Unlike depth or infrared sensors, RGB cameras are ubiquitous in smartphones and standard car equipment; (2) Sufficient detail for many applications: RGB data often provide enough visual information to detect various forms of driver distraction, such as looking away from the road, using a phone, or falling asleep. The color and texture details captured in RGB images are usually sufficient for these purposes; (3) Computational efficiency: processing RGB data typically requires less computational power compared to methods that also involve depth or infrared data. This makes the RGB-based method more suitable for real-time applications where rapid processing is crucial; and finally (4) Broader research and development: there is a wealth of research and development in the field of computer vision using RGB data, leading to more advanced and refined algorithms for distraction detection.

Furthermore, as autonomous driving systems need a fast response and accurate results, the presented paper focuses on the related work which was established based on the MobileNet model and evaluated using the State-Farm dataset, such as in [12–14].

In [12], the MobileNetV2 is used by the authors to present a CNN-based methodology for detecting and determining the cause of distracted driving. MobileNetV2 is used as the backbone network in the model, with a global average pooling layer, a dropout layer, and a fully connected layer added on top. The model is also evaluated using several metrics such as accuracy, precision, recall, F1-score, and the confusion matrix. According to the study, the suggested MobileNetV2 model has the greatest accuracy of 98.12% and the lowest loss of 0.0937 across all models. According to the paper, MobileNetV2 is the ideal solution for resource-constrained devices like mobile and embedded vision applications. According to the study, the suggested model can identify driver attention and offer real-time feedback to avert accidents.

Also, the authors of [13] used the MobileNetV2 model through an evaluation comparison for the proposed system. According to the study, the suggested system monitors the driver's behavior and identifies distractions to inform them. The system employs a convolutional neural network (CNN) model to categorize driver activity into ten separate classifications, nine of which include the driver being distracted by other activities and one being "safe driving". The device captures photos of the driver's face and upper torso using a camera installed on the dashboard. The authors use the MobileNetV2 architecture as one of the transfer learning models that are used to evaluate the proposed model. The results state that the MobileNetV2 model achieves the lowest performance of 89.2% as an accuracy value, the proposed system has the highest accuracy of 93.9%, the DenseNet obtained an accuracy of 92.4%, and Inception-v3 obtained an accuracy of 90.8%. According

to the conclusion, by informing drivers in real-time, the device can successfully minimize road accidents caused by driver distraction. The paper also suggests some future system improvements, such as adding temporal context to capture the dynamics of driver behavior, using steering wheel vibrations to provide haptic feedback to drivers, reducing car speed during distractions to avoid collisions, and addressing other types of distractions, such as cognitive and auditory distractions.

Furthermore, in [14], the MobileNetV2 is used throughout the research study to make a comprehensive comparison of the proposed technique. The research offers a system that detects and alerts inattentive drivers using a deep-learning ensemble model. The technology also gives advice to drivers to reduce distractions and boost in-car awareness for greater safety. Throughout the paper, the MobileNetV2 model is used separately, which achieved an accuracy of 82%. Also, it is combined with other models (e.g., Inception, VGG16, and ResNet50) in the proposed stacked ensemble technique to increase overall accuracy (e.g., the MobileNet-Inception variation, which obtained an accuracy of 88%). The article finds that by notifying drivers in real-time, the technology can successfully avoid road accidents caused by driver attention. However, the authors highlight the computational difficulty stemming from the integration of big models.

1.2. The Proposed Contributions

The work proposed in this paper makes the following contributions:

- Demonstrating the relationship between autonomous driving and distracted driver detection, with a declaration of the challenges through performing the distracted driver detection task. In addition, a review of the papers that perform this task by using the MobileNet model and the State-Farm dataset.
- Proposing a deep learning architecture, based on the MobileNet model as a backbone for the architecture, with a declaration of the specific reasons behind the selection of this model.
- Making a comparison between the performance of the proposed work with respect to the previous papers that were performed using the same data and the MobileNet architecture. This comparison declares that the proposed work outperforms most of the previous work.

1.3. The Proposed Paper Organization

This paper contains four sections, besides the introduction section (Section 1), that declare the motivation for this paper, review the related work, and outline the contribution of the present paper, respectively. In Section 2, namely the methodology section, the authors demonstrate the architecture of the proposed work, the details of the used data, the way for the performance evaluation, and the required setups for the architecture training. For the visualization of the results and the discussion of the performance, the paper contains Section 3. In Section 4, the limitation and expected future work are demonstrated. Finally, Section 5 contained the conclusions of this paper.

Table 3. Comparison between the previous works that use MobileNet architecture with the State-Farm dataset and the proposed work.

Ref.	Architecture	Architecture Details per Layer	Architecture Parameters	Loss	Optimizer	Augmentation	Prediction Time (Seconds)	Recall (%)	Precision (%)	F1-Score (%)	Loss	Accuracy (%)
[12]	• MobileNetV2 • Global average pooling • Dropout • Dense	Not stated	3.5×10^6	Categorical cross-entropy	Adam	Rotation, shifting, scaling, flipping, and brightness	Not stated	Not stated	Not stated	Not stated	0.0937	98.12
[13]	• MobileNetV2		2,257,984			Horizontal flipping, rotation, zooming, and shifting					Not stated	89.2
[14]	• MobileNetV2 • Global average pooling • Two dense • Batch normalization • Dropout	Stated	4,812,490			Horizontal flipping, rotation, zooming, and shifting		82	83	82	0.5	82
Ours	• MobileNetV1 • Global average pooling • Three dense • Batch normalization • Three dropout	Stated	4,939,210		SGD	Rotation, zooming, and shifting	0.01	88.8	90.7	89.8	0.3652	89.63

2. Methodology

2.1. Dataset

Earlier datasets concentrate on a small number of distractions, many of which are no longer publicly available. State-Farm's distracted driver detection competition on Kaggle defined 10 postures to be discovered [15]. This was the first set of statistics to examine a wide range of distractions and was made public.

The State-Farm dataset mainly contains two folders, one for the training dataset and the other for the testing dataset. There are 22,424 images in the dataset's training folder that have been labeled. The data that are in the testing folder comprise 29,700 unlabeled images. As a result, photographs from the training folder, which contained 22,424 images, were used in this investigation. The collection includes ten distinct types of driver problems, that are named from C0 to C9 as class names. Table 2 displays the picture attributes included in the dataset's image classes.

In Table 2, it is clear that the data have a class imbalance in terms of the number of images per each class. This is especially the case for the C8 class (hair and makeup class), which contains only 1911 images, while the C0 class (safe driving class) contains 2489 images. Hereby, during the proposed architecture training, the augmentation technique is used to overcome the class imbalance by making transformation processes like shifting, rotation, and zooming. Also, the data augmentation helps in producing a robust trained model during deployment.

2.2. Proposed Architecture

2.2.1. Main Architecture Components

MobileNet is the first TensorFlow-based mobile deep learning model. Because its design and computational cost are substantially simpler when compared to other transfer learning models (e.g., VGG and ResNet), the term Mobile suggests that the model may work in mobile applications. MobileNet is built on a method known as separable depth-wise convolutions, which greatly decreases the computational cost by lowering the number of parameters, particularly when compared to networks using the standard convolutions of the same depth. MobileNet is therefore a lightweight deep neural network appropriate for mobile applications. The technique of depth-wise separable convolution is created by combining two fundamental operations: depth-wise convolution and point-wise convolution. The concept behind depth-wise convolution was that the spatial and depth aspects of the filter might be separated. The height and width dimensions of the filter are separated, and then the depth dimension is separated from the horizontal (width $\times$ height) dimension. The point-wise convolution is a 1×1 convolution that modifies the preceding layer's dimension [14,22,23]. This transfer learning model is used as a backbone for the overall architecture, to perform feature extraction from the human driver images with a size of ($224 \times 224 \times 3$).

Global average pooling 2D is the layer that averages all values per channel to convert the 2D channel into only one value. This layer helps reduce the number of parameters and computational costs for the model [24]. This layer is inserted after the MobileNet model to convert the extracted features from ($7 \times 7 \times 1024$) into (1×1024), by averaging the values of each (7×7) channel.

Dense is one of the most basic and widely used layers in deep learning models because it is a core layer that implements a fully connected neural network layer. It can learn complex patterns and representations from the input data by performing a transformation followed by an activation function [25]. This layer type is used many times after the global average layer, until obtaining the final classification output.

Batch normalization is a layer that uses the mean and standard deviation of the current batch during training to normalize its inputs. In addition, it assists in the prevention of overfitting, the acceleration of training, and the reduction in the model's reliance on the initialization and learning rates [26]. The batch normalization layer is used after one of the dense layers to make use of its advantages.

Dropout is a regularization layer that randomly changes part of the input units to zero during training to prevent overfitting. This change is performed based on a fraction called rate. The rate, which is the proportion of units to drop, is scaled up by the dropout layer, which increases the remaining units by a factor of 1/(1-rate) [27]. The dropout layer is inserted before some dense layers, to perform regularization and avoid overfitting.

2.2.2. Overall Proposed Architecture Layout

As shown in Figure 1, the proposed architecture starts with the MobileNet model as a backbone to perform the feature extraction process by taking RGB images as input of size ($224 \times 224 \times 3$) and extracting features as an output of size ($7 \times 7 \times 1024$). Then, the global average pooling layer converts the extracted features into a size of (1×1024) by making an averaging process per each extracted feature channel. Hereby, the dense layer with 1024 units and ReLU activation function is inserted, followed by a dropout layer with a rate value of 0.5. Therefore, a new dense layer with 512 units and a ReLU activation function is added, followed by a batch normalization layer and a dropout layer with a rate value of 0.3. Furthermore, a further 256 units of a dense layer with a ReLU activation function are appended, followed by a dropout layer with a rate value of 0.1.

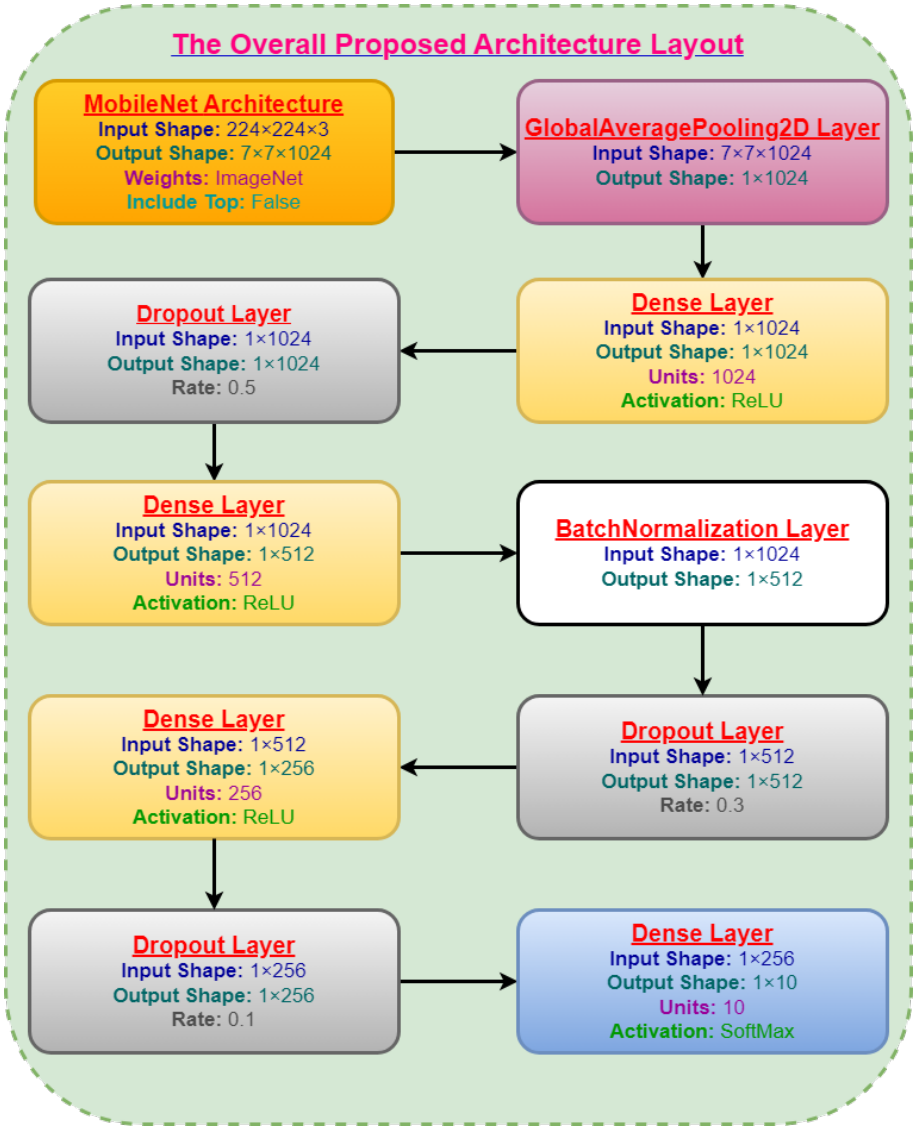

Figure 1. The proposed architecture layout in the details of each layer.

Finally, a dense layer with 10 units (i.e., one output per each driver case class) and a softmax activation function, are added to extract the final probability per each class. The hyper-parameters for the proposed architecture are adjusted based on the main author's experience of handling such cases over the course of many papers [28–31].

2.3. Evaluation Metrics

Accuracy, recall, precision, f1-score, and confusion matrix are the common metrics that have been used for evaluating the multi-class classifiers. These are based on the concepts of true positives (TPs), false positives (FPs), and false negatives (FNs). A true positive is an instance that is projected to belong to a class. A false positive is a case in which an instance does not belong to a class but is projected to do so. True negatives are instances that do not belong to a class and are not projected to be so. A false negative is an instance that belongs to a class but is not projected to be a member of that class.

Accuracy (as shown in Equation (1)) is a measure of how successfully a multi-class classifier predicts the input data's class labels. Recall (as shown in Equation (2)) is the proportion of actual positive instances that are correctly predicted by the classifier. Precision (as shown in Equation (3)) is the proportion of predicted positive instances that are actually positive. F1-score (as shown in Equation (4)) is the combination of recall and precision metrics which serves to identify how effectively the model can detect real positives among all positive instances and predictions. A confusion matrix, which is a table that gives the number of TPs, FPs, TNs, and FNs for each class, is required to determine the accuracy of multi-class classification.

$$\text{Accuracy} = \frac{\text{True Positive + True Negative}}{\text{True Positive + True Negative + False Positive + False Negative}} \quad (1)$$

$$\text{Recall} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}} \quad (2)$$

$$\text{Precision} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}} \quad (3)$$

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (4)$$

2.4. Experimental Setup

The suggested architecture is developed and evaluated using the Python programming language's TensorFlow library using a computer that has a Windows 10 operating system, and hardware specifications of (a 12th Gen Intel(R) Core(TM) i9-12900KF 3.20 GHz) processor, (32.0 GB) RAM size, and (NVIDIA GeForce RTX 3090-Ti) GPU.

After ingesting the Distracted Driver dataset, the following preprocessing is performed: (1) The driver (_imgs_list.csv) file and the images are loaded; (2) The images are loaded as RGB and resized to ($244 \times 244 \times 3$) because this is the MobileNet model's input size; (3) The data are divided into training and validation sets. The training and validation sets were created based on the subject (driver ID), the subjects selected for the validation were p015, p022, p050, and p056 (as mentioned in [14]), while IDs of the remaining drivers are kept as a training set; and finally (4) The classification labels are transformed into category values using the one hot encoder. Consequently, there were 13,770 photos in the training set and 3692 in the validation set.

For the model training, stochastic gradient descent (SGD) is selected as an optimizer; with a learning rate value of 0.005, and a momentum value of 0.5. Furthermore, the categorical cross-entropy is selected as a loss function for the distracted driver detection model training. Also, some transformations are performed by an image data generator for the training images through the training time (e.g., height and width shifting with values of 0.5, zooming with 0.5 value, and rotation with 30 degrees). Furthermore, the maximum

number of epochs for the training is set to 100 epochs. At the same time, the early stopping technique is used during the proposed architecture training process that monitors the validation loss value and stops the training before the performance degradation.

3. Results and Discussion

First of all, by looking into the proposed architecture training curves (as shown in Figure 2), it is clear that the curves converge to the best validation data performance after around six epochs. After that, the model starts to overfit the data, so the best weights during the training process are saved using a callback function based on the validation loss monitoring. With this, the best model achieves a training accuracy of 87.75% and a validation accuracy of 89.63%. Also, the model obtains a training loss of 0.3966 and a validation loss of 0.3652. Then, the performance curves start to overfit the training data, so the validation curves yield worse results while the training curves progress towards yielding better values.

Furthermore, by looking much more deeply into the model performance through the visualization for the normalized confusion matrix (as shown in Figure 3), most of the classes are correctly classified by the proposed architecture (e.g., exiting, talking on the phone, and reaching behind classes). On the other hand, there are two classes for which the results of the proposed model are fuzzy, the first one being the safe driving class that is sometimes incorrectly classified as a talking to passenger class, and the second one being the talking to passenger class that may be incorrectly classified as a safe driving class or a hair and makeup class. However, the overall architecture performance overcomes this issue.

Hereby, the proposed architecture achieves an overall acceptable performance (i.e., with training and validation accuracies of 87.75% and 89.63%, respectively; and training and validation losses of 0.3966 and 0.3652, respectively), especially when compared with the previous work in Table 3. In addition, the average processing time for the architecture to make the classification process is 0.01 seconds per image, which means a high response time for the driver case identification. In other words, the fast response for the driver case decreases the probability of there being an accident by alerting the driver much faster in the emergency situations that the autonomous system cannot handle. Furthermore, an implicit performance evaluation is performed based on the recall, precision, and f1-score values. This evaluation resulted in a validation recall value of 88.8%, a validation precision value of 90.7%, and a validation f1-score value of 89.8%. These results can additionally show the proposed architecture's effectiveness. Also, by looking into Table 3, it can be noted that the proposed work is more informative compared to the related works that do not contain details about the architecture or the evaluation.

In Table 3, the contribution of the proposed work compared to the previous works is declared by making a comparison in terms of the architecture components, the explanation for architecture details for each layer, the number of parameters or weights for each architecture, the loss function, the optimizer, the augmentation techniques, and the evaluation metrics such as the average prediction time, recall, precision, f1-score, loss, and accuracy. For the architecture components, all previous works are constructed based on the MobileNetV2 model, while the proposed architecture is based on MobileNetV1 architecture. For the declaration of architecture details (e.g., the number of units per layer, dropout rate values, and the type of activation function); the proposed work is fully informative compared to the previous works that rarely describe any details as in [12,13]. For the number of parameters in an architecture, the proposed architecture has a slightly higher number than the one in [14], while the proposed work achieves a better performance. The loss function has been applied identically in all works (i.e., the categorical cross-entropy function). The Adam optimizer has been leveraged in previous works, while the proposed work uses the SGD optimizer. For the augmentation techniques, the proposed work only uses three transformation techniques, while the previous works use more than that. For the performance evaluation, the proposed architecture is evaluated using various criteria (e.g., the average prediction time, recall, precision, f1-score, loss, and accuracy), unlike previous works such as [12–14].

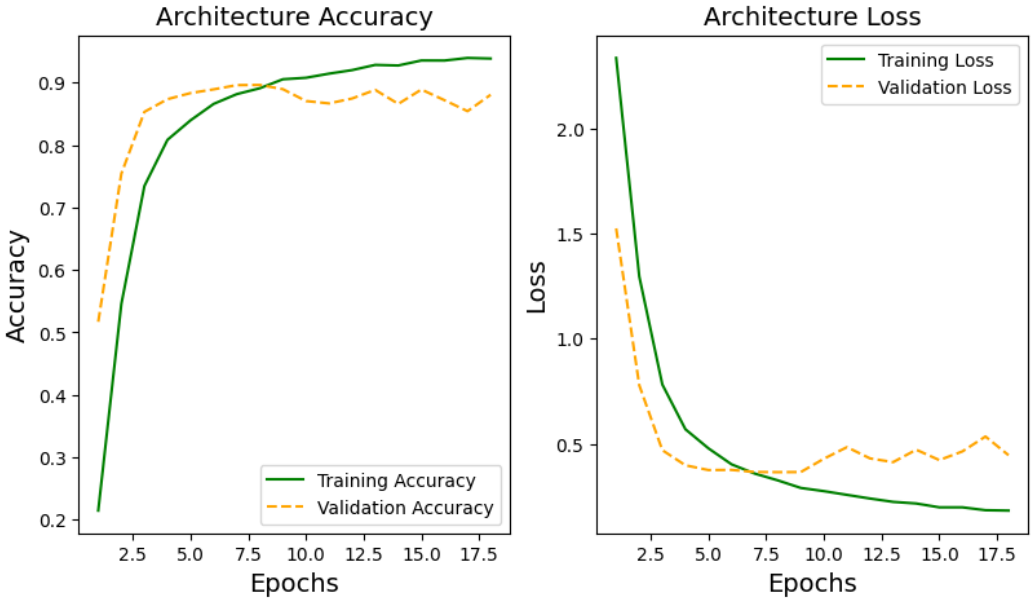

Figure 2. The accuracy and loss curves for the proposed architecture through training and validation epochs.

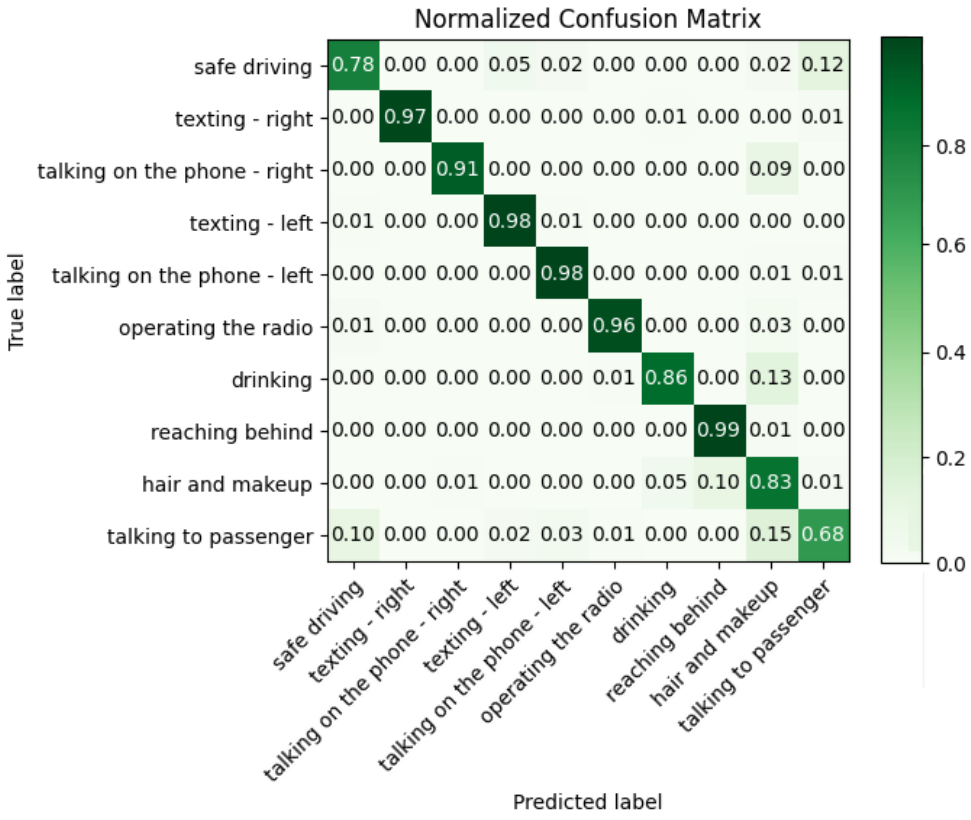

Figure 3. The confusion matrix for the proposed architecture using the validation dataset.

For more results, the declaration of the ten human driver conditions stated in the State-Farm dataset and predicted by the proposed architecture, the visualization of some correct prediction examples and some wrong prediction examples (as shown in Figure 4) from the validation dataset are required.

Figure 4. Examples of the correct and incorrect prediction results of the proposed architecture for the ten human driver conditions according to the State-Farm dataset, in which the TL is the true label and the PL is the predicted label.

4. Limitations and Future Work

Actually, the limitations of the proposed architecture is directly correlated with expected future work. However, the results and discussion section demonstrates the high performance of the proposed architecture. At the same time, the section mentions the misclassification problem for two specific classes (i.e., safe driving and talking to passenger classes). This issue is the main limitation of the proposed architecture because it degrades the overall performance, as clarified through the confusion matrix visualization. Hence, the expected future work will be totally focused by increasing the classification accuracy for the specified classes (i.e., safe driving and talking to passengers) and making some additional helpful techniques (e.g., ensampling, and features extraction techniques). Hereby, the overall classification model performance will be enhanced.

In addition, the results and discussion section provides the fast response for the proposed architecture. Therefore, the expected future work may study the implementation of

the architecture in a real-time system with hardware components to prove the applicability of the proposed architecture.

5. Conclusions

This study introduces a sufficient architecture that effectively utilizes the MobileNet transfer learning model as its backbone in order to classify ten types of human driver distraction as featured in the State-Farm dataset. Through proposing this architecture, the paper is organized based on three main points:

Firstly, to declare the importance of distracted human driver detection in real-life and autonomous driving, the paper starts by explaining the interaction between autonomous driving cars and the distracted driver detection task with a declaration of the autonomous driving target levels for this task and the task challenges (i.e., the fast response time and high-performance results) with a demonstration of the importance of distracted human driver detection in real-life applications.

Secondly, to demonstrate the proposed architecture's effectiveness and efficiency, the paper reviews the related work which works on a MobileNet-based architecture and uses the State-Farm dataset and demonstrates the reason for selecting the MobileNet model as a backbone for this architecture, which serves to obtain a fast response for the detection of the human driver condition. Also, the proposed architecture is declared in detail (which is not provided in most of the previous work stated in the review part). In addition, this paper presents a comparative evaluation of the proposed architecture based on the accuracy, recall, precision, f1-score, the confusion matrix, and the response time, with a declaration of the importance of the confusion matrix through the analysis of the results. Furthermore, the performances of the proposed architecture and the previous work reviewed herein are compared to state that the proposed work outperforms most of the related work because it achieves a validation accuracy of 89.63%, a validation recall of 88.8%, a validation precision of 90.7%, a validation f1-score of 89.8%, a validation loss of 0.3652, and a prediction time of about 0.01 seconds per image.

Finally, to declare the limitations and the expected future work for the proposed architecture, this paper explains the classification imbalance limitation of the proposed architecture, and the expected future work of this paper to enhance this problem by making some additional helpful techniques (e.g., ensampling and features extraction techniques). Hereby, this elaborates a real-time hardware system to perform the task in real life.

Author Contributions: Conceptualization, M.A.B.A.; methodology, M.A.B.A.; software, M.A.B.A.; validation, M.A.B.A.; formal analysis, M.A.B.A.; investigation, M.A.B.A.; resources, M.A.B.A.; data curation, M.A.B.A.; writing—original draft preparation, M.A.B.A.; writing—review and editing, M.A.B.A. and Y.B.; visualization, M.A.B.A.; supervision, Y.B.; project administration, Y.B.; funding acquisition, Y.B. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korean government (Ministry of Science and ICT) (No. 2022R1A5A8026986 and No. 2022R1F1A1073745), the Korea Institute for Advancement of Technology (KIAT) grant funded by the Korea Government (MOTIE) (P0020536, HRD Program for Industrial Innovation), and the Chungbuk National University BK21 program (2021).

Data Availability Statement: The data used herein Are from State-Farm's distracted driver detection competition on Kaggle at https://www.kaggle.com/competitions/state-farm-distracted-driver-detection/data (accessed on 17 November 2023).

Acknowledgments: Special thanks to the Ministry of Science and ICT(Korean government), Ministry of Trade, Industry and Energy(Korean government), and Chungbuk National University for providing overall support.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

BASt	German Federal Highway Research Institute
CNN	Convolutional Neural Network
FN	False Negative
FP	False Positive
NHTSA	US National Highway Traffic Safety Administration
OICA	International Organization of Motor Vehicle Manufacturers
PL	Predicted Label
SAE	Society of Automotive Engineers
SGD	Stochastic Gradient Descent
TL	True Label
TN	True Negative
TP	True Positive

References

1. WHO. *Road Traffic Injuries*; WHO: Geneva, Switzerland, 2020.
2. Yanbin, Y.; Lijuan, Z.; Mengjun, L.; Ling, S. Early warning of traffic accident in Shanghai based on large data set mining. In Proceedings of the 2016 International Conference on Intelligent Transportation, Big Data & Smart City (ICITBS), Changsha, China, 17–18 December 2016; pp. 18–21.
3. Park, K.; Im, Y. Ergonomic Guidelines of Head-Up Display User Interface during Semi-Automated Driving. *Electronics* **2020**, *9*, 611. [CrossRef]
4. Ledezma, A.; Zamora, V.; Sipele, O.; Sesmero, M.P.; Sanchis, A. Implementing a Gaze Tracking Algorithm for Improving Advanced Driver Assistance Systems. *Electronics* **2021**, *10*, 1480. [CrossRef]
5. Han, J.H.; Ju, D.Y. Advanced Alarm Method Based on Driver's State in Autonomous Vehicles. *Electronics* **2021**, *10*, 2796. [CrossRef]
6. Li, T.; Chang, X.; Wu, Z.; Li, J.; Shao, G.; Deng, X.; Qiu, J.; Guo, B.; Zhang, G.; He, Q.; et al. Autonomous Collision-Free Navigation of Microvehicles in Complex and Dynamically Changing Environments. *ACS Nano* **2017**, *11*, 9268–9275. [CrossRef] [PubMed]
7. Zanchin, B.C.; Adamshuk, R.; Santos, M.M.; Collazos, K.S. On the instrumentation and classification of autonomous cars. In Proceedings of the 2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Banff, AB, Canada, 5–8 October 2017; pp. 2631–2636.
8. Ahangar, M.N.; Ahmed, Q.Z.; Khan, F.A.; Hafeez, M. A Survey of Autonomous Vehicles: Enabling Communication Technologies and Challenges. *Sensors* **2021**, *21*, 706. [CrossRef]
9. Shahian Jahromi, B.; Hussain, S.; Karakas, B.; Cetin, S. Control of autonomous ground vehicles: A brief technical review. *Iop Conf. Ser. Mater. Sci. Eng.* **2017**, *224*, 012029. [CrossRef]
10. Flores-Monroy, J.; Nakano-Miyatake, M.; Escamilla-Hernandez, E.; Sanchez-Perez, G.; Perez-Meana, H. SOMN_IA: Portable and Universal Device for Real-Time Detection of Driver's Drowsiness and Distraction Levels. *Electronics* **2022**, *11*, 2558. [CrossRef]
11. Anber, S.; Alsaggaf, W.; Shalash, W. A Hybrid Driver Fatigue and Distraction Detection Model Using AlexNet Based on Facial Features. *Electronics* **2022**, *11*, 285. [CrossRef]
12. Hossain, M.U.; Rahman, M.A.; Islam, M.M.; Akhter, A.; Uddin, M.A.; Paul, B.K. Automatic driver distraction detection using deep convolutional neural networks. *Intell. Syst. Appl.* **2022**, *14*, 200075. [CrossRef]
13. Pal, A.; Kar, S.; Bharti, M. Algorithm for Distracted Driver Detection and Alert Using Deep Learning. *Opt. Mem. Neural Netw.* **2021**, *30*, 257–265.
14. Aljasim, M.; Kashef, R. E2DR: A Deep Learning Ensemble-Based Driver Distraction Detection with Recommendations Model. *Sensors* **2022**, *22*, 1858. [CrossRef] [PubMed]
15. Montoya, A.; Holman, D. State Farm Distracted Driver Detection. *Int. J. Eng. Res. Appl.* **2016**, *4*, 123.
16. Abouelnaga, Y.; Eraqi, H.M.; Moustafa, M.N. Real-time Distracted Driver Posture Classification. *arXiv* **2018**, arXiv:1706.09498.
17. Jain, A.; Koppula, H.S.; Raghavan, B.; Soh, S.; Saxena, A. Car that Knows Before You Do: Anticipating Maneuvers via Learning Temporal Driving Models. *arXiv* **2015**, arXiv:1504.02789.
18. Ohn-Bar, E.; Martin, S.; Tawari, A.; Trivedi, M.M. Head, Eye, and Hand Patterns for Driver Activity Recognition. In Proceedings of the 2014 22nd International Conference on Pattern Recognition, Stockholm, Sweden, 24–28 August 2014; pp. 660–665.
19. Cruz, S.D.D.; Wasenmüller, O.; Beise, H.P.; Stifter, T.; Stricker, D. SVIRO: Synthetic Vehicle Interior Rear Seat Occupancy Dataset and Benchmark. *arXiv* **2020**, arXiv:2001.03483.
20. Martin, M.; Roitberg, A.; Haurilet, M.; Horne, M.; Reiss, S.; Voit, M.; Stiefelhagen, R. Drive&Act: A Multi-Modal Dataset for Fine-Grained Driver Behavior Recognition in Autonomous Vehicles. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27 October–2 November 2019.
21. Katrolia, J.S.; Mirbach, B.; El-Sherif, A.; Feld, H.; Rambach, J.; Stricker, D. TICaM: A Time-of-flight In-car Cabin Monitoring Dataset. *arXiv* **2021**, arXiv:2103.11719.

22. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv* **2017**, arXiv:1704.04861.
23. Kim, W.; Jung, W.S.; Choi, H.K. Lightweight driver monitoring system based on multi-task mobilenets. *Sensors* **2019**, *19*, 3200. [CrossRef]
24. Lin, M.; Chen, Q.; Yan, S. Network In Network. *arXiv* **2014**, arXiv:1312.4400.
25. Chollet, F. *Deep Learning with Python*; Manning Publications: Shelter Island, NY, USA, 2017.
26. Ioffe, S.; Szegedy, C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. *arXiv* **2015**, arXiv:1502.03167.
27. Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *J. Mach. Learn. Res.* **2014**, *15*, 1929–1958.
28. Abbass, M.A.B.; Kang, H.S. Violence Detection Enhancement by Involving Convolutional Block Attention Modules Into Various Deep Learning Architectures: Comprehensive Case Study for UBI-Fights Dataset. *IEEE Access* **2023**, *11*, 37096–37107. [CrossRef]
29. Abbass, M.A.B.; Hamdy, M. A Generic Pipeline for Machine Learning Users in Energy and Buildings Domain. *Energies* **2021**, *14*, 5410. [CrossRef]
30. Abbass, M.A.B.; Kang, H.S. Drone Elevation Control Based on Python-Unity Integrated Framework for Reinforcement Learning Applications. *Drones* **2023**, *7*, 225. [CrossRef]
31. Abbass, M.A.B. A comprehensive framework based on Bayesian optimization and skip connections artificial neural networks to predict buildings energy performance. *J. Build. Eng.* **2023**, *77*, 107523. [CrossRef]