Penerbit Akademia Baru

Journal of Advanced Research Design

Journal homepage:
https://akademiabaru.com/submit/index.php/ard
ISSN: 2289-7984

An Efficient Distracted Driving Detection Based on MobileNet V2SE Fusion

Lee Chin Kho[1,*], Guo Cheng[2], Sze Song Ngu[1], Qi Zhe Koh[1], Annie Joseph[1], Kuryati Kipli[1]

[1] Department of Electrical and Electronics Engineering, Faculty of Engineering, Universiti Malaysia Sarawak (UNIMAS), Kota Samarahan, Sarawak, Malaysia
[2] Faculty of Computer Science and Technology, University Bengbu, Bengbu, China

ARTICLE INFO

Article history:
Received 27 March 2025
Received in revised form 18 August 2025
Accepted 29 September 2025
Available online 10 October 2025

Keywords:
Deep learning; lightweight; distracted driving; accuracy; squeeze and excite (SE)

ABSTRACT

The issue of distracted driving has become a significant concern, leading to numerous fatalities and injuries. There is a pressing need to develop innovative approaches to identify and mitigate this problem. This paper proposes a lightweight deep learning model that uses MobileNetV2 as the base and includes attention mechanisms like the Squeeze and Excite (SE) module to identify distracted driver actions. The proposed model underwent rigorous training and testing using the American University in Cairo (AUC) distracted driver dataset, which includes ten distraction categories. The model was optimized through hyperparameter tuning, data augmentation, and class weighting. To validate the model's effectiveness, a confusion matrix, frames per second (FPS), accuracy, precision, recall, and F1 score were used as evaluation metrics. The proposed model achieved 93% accuracy with a batch size of 32, learning rate of 0.0001, and 21 epochs. Furthermore, the proposed model was compared to the MobileNetV2 and other existing architectures regarding accuracy and parameters. The proposed method outperformed unmodified deep learning models and maintained a balance between accuracy and parameter utilization, while some other modified models performed slightly better. The proposed method shows promising potential for accurately detecting distracted drivers with efficiency.

1. Introduction

Every year, approximately 30 to 50 million people suffer injuries, and about 1.3 million people die due to traffic accidents, according to the World Health Organization's statistics [1]. While many factors contribute to these numbers, distracted driving is the primary one. The National Highway Traffic Safety Administration (NHTSA) defines distracted driving as any activity that takes the driver's attention away from driving, such as using electronic devices, eating or drinking, talking with passengers, adjusting in-car technology, or engaging in other activities [2].

Driver distraction can occur in three different ways: visual, manual, and cognitive. Visual distractions happen when drivers take their eyes off the road, while manual distractions occur when they remove their hands from the steering wheel. Cognitive distractions occur when a driver's mind

[*] *Corresponding Author.*
E-mail address: lckho@unimas.my

https://doi.org/10.37934/ard.145.1.149164

is not fully focused on driving [3]. Drivers should be cautious of visual distractions since they can cause drivers to miss important visual cues like traffic signals, pedestrians, or other vehicles.

Manual distractions, such as eating or changing the radio station, can make a driver lose control of their vehicle. Drivers face more distractions with the widespread use of smartphones and navigation systems. Some car manufacturers have added advanced infotainment systems, control interfaces, and display systems to their cars, which can be particularly distracting while driving. Cognitive distractions occur when a driver is not mentally focused on driving due to reasons like stress, fatigue, daydreaming, personal problems, or engaging in an intense conversation. This type of distraction can be dangerous since it can cause drivers to miss seeing important things on the road or react slowly in an emergency situation.

It's important to be aware of driver distractions, which have become a critical concern and contribute to numerous accidents, endangering road life. Traditional methods for identifying and mitigating distractions while driving have limitations. However, with the introduction of deep learning, new approaches have opened up to tackle this issue effectively. The development of wearable technology with integrated sensors has made it possible to monitor the driver's physiological state, including electroencephalogram and electrocardiogram. Compared to behavioral and visual approaches, this method is more reliable. However, these devices can cause discomfort and restrict movement, and they can also be susceptible to interference caused by bodily phenomena like direct contact with the skin [4].

Researchers have developed various methods for recognizing distracted driving behavior based on deep learning. However, the cost of computing has become a challenge for model edge-oriented migration, and most automobiles' onboard electronics have significantly less powerful microcontrollers than mobile phones. Therefore, this paper focuses on two main problems: the inability to deploy trained models from traditional networks on edge devices and the relatively lower accuracy of lightweight networks. The model must be compact, efficient, and have fast processing time due to the restricted processing power of the edge devices in the car. The MobileNetV2 network, with a total of 3,500,000 parameters, is an optimal choice for this application. It is a lightweight network that is sufficiently robust to handle the American University in Cairo (AUC) Distracted Driver Dataset.

The paper proposes a deep-learning lightweight model based on MobileNetV2 to detect distracted driving actions. The proposed model aims to optimize the performance by adding a Squeeze and Excitation (SE) block and modifying the layers of MobileNetV2, becoming the name of MobileNetV2SE fusion. The study compares the accuracy and processing time of the proposed method with other models. The proposed method utilizes MobileNetV2, a lightweight architecture designed for mobile and embedded devices, to identify patterns of distracted driving and caution drivers to refocus on the road with satisfactory accuracy and processing time. The study uses the America University in Cairo (AUC) dataset to train the model, which contains 31 drivers of various genders and races from seven different nations, giving it a significant advantage over other datasets. In summary, the paper's contribution includes (i) proposing a new model based on MobileNetV2 to detect distracted driving, (ii) demonstrating its effectiveness by comparing it with other models, and (iii) providing a better way to detect risky driving behaviors, thus contributing to safer roads.

The rest of this paper is organized as follows: Section 2 discusses the related work, while Section 3 explains the steps taken to improve the existing MobileNetV2 network. The results are illustrated in Section 4, and the conclusion is presented in Section 5.

2. Related Work

There are various methods to identify signs of driver distraction, with machine learning and deep learning being two popular approaches. These subfields of artificial intelligence teach algorithms how to learn from data and make predictions without explicit programming. Machine learning focuses on pattern recognition and decision-making, which is divided into three categories: supervised, unsupervised, and reinforcement learning. Deep learning, a subset of machine learning, mimics the human brain using artificial neural networks. It is particularly adept at learning hierarchical representations from complex, unstructured data such as images, text, and audio. Convolutional Neural Networks (CNNs) are commonly used for image recognition tasks.

2.1 Machine learning approaches

A developed thesis utilizes machine learning techniques such as linear support vector machines (SVM), softmax, naïve Bayes, decision trees, and a 2-layer neural network to effectively identify distracted driving behaviors from images of drivers inside their vehicles [5]. The study achieved an impressive 92.24% accuracy rate, demonstrating the exceptional capability of the model to detect distracted driving incidents accurately. In another paper, the authors propose an innovative method that uses fuzzy set theory to evaluate driver distraction and situational awareness while performing secondary tasks [5]. The proposed method utilizes a rule-based fuzzy logic system that considers various factors, such as lane keeping and speed limit adherence, and can accurately identify and quantify driver distraction levels as a percentage based on safe vehicle dynamic performance. The proposed method is more accurate than previous laboratory-based approaches due to the inclusion of additional input measures. Besides, a study conducted analyzed seven different machine learning models, including SVM, decision tree (DT), logistic regression (LR), random forest (RF), AdaBoost (ADB), gradient boosting (GB), and CNN to distinguish between drivers and passengers [6]. The study concludes that the CNN and GB models are highly effective in accurately distinguishing between the two.

2.2 Deep learning approaches

When researching driving distraction, the most commonly used technique is the CNN, explicitly designed for image processing and recognition. The method used in most studies to detect driver attention differs depending on the type of distraction, such as cognitive, manual, and visual distraction. Deep learning approaches in distracted driving can be divided into three categories: heavy CNN models, lightweight CNN models, and lightweight CNN models combined with other techniques.

2.2.1 Heavy CNN models

Researchers have developed different methods to identify driver distraction using deep learning technology. Ezzouhri *et al.*, [7] employed two CNN-based classification models, Visual Geometry Group 19 (VGG-19) and Inception-V3, to identify drivers' distracted activities and the Cross Domain Complementary Learning (CDCL) architecture was used to segment critical body parts. This approach achieved an average accuracy of over 96% on one dataset and 95% on another. In [8], the authors used Deep Learning-based categorization with the Residual Network-50 (ResNet-50) network to identify distracted drivers gazing elsewhere using the StateFarm dataset. This method achieved an accuracy of 94%. The output generated by this method distinguished between different types of distracted behaviour using red and green colours. In [9], ResNet-50's deep features were merged with the SVM classifier to create ReSVM. ReSVM employs the deep features of the last multilayer perceptron convolution layer, which are derived from input photos of various sizes and lighting conditions. The SVM layer is then given the computed mean of the feature map for classification. This approach achieved up to 95.5% accuracy on four different datasets.

Besides, a unique hybrid method was introduced in [10] that uses stacked Bidirectional Long Short-Term Memory (BiLSTM) Networks and a pre-trained convolutional neural network (CNN) architecture, InceptionV3, to capture the spectral-spatial characteristics of images. This method achieved an average accuracy of 92.7% on the AUC dataset. In [11], the authors proposed a deep learning framework for detecting distracted drivers using a pre-trained model, EfficientNet. The model utilizes a technique for analyzing images and identifying regions of interest (ROI) related to body parts and objects involved in distracting activities. Specifically, it employs five variations of the Efficientdet model (D0-D4) for detection purposes. Lastly, the cognitive feature of distracted driver behavior caused by the driver's drowsiness was determined by detecting the sequential blinking of the eyes [12]. The first steps are the blink detection and feature extraction from the Real-life drowsiness (RLDD) dataset. They structured drowsiness detection as a regression problem. Then, they incorporated a Hierarchical Multiscale LSTM (HMLSTM) module into the model. The regression output was discretized to produce a categorization label for each video segment. The voting process was applied on top of the categorization outcomes for every video segment. However, the downside of this method is that the dataset was not collected under driving conditions, and it is challenging to observe blinking if the video is at low frame rates.

Most of the models used in studies are highly accurate, with a 90% success rate. However, this accuracy comes at the cost of computational complexity, making them unsuitable for real-time implementation and embedded deployment. To ensure a timely and efficient deployment, it is essential to develop distraction detection models with fewer parameters while maintaining their effectiveness.

2.2.2 Lightweight CNN models

There are mainly two approaches to creating a lightweight CNN. The first method involves compressing the existing network, which can be highly effective. This technique involves resizing the convolution kernels, freezing the convolution layers, and fully connecting later. The second approach is to build a new network module. Currently, most of the research is focused on the convolution operation.

Abouelnaga *et al.,* [13] proposed a system to identify distracted driving postures using face and hand detection. They used two pre-trained neural networks, InceptionV3 and AlexNet, for feature extraction and classification. However, they found a more straightforward model using only AlexNet, which maintained high accuracy for real-time applications. In [14], an innovative method was developed to identify distracted driving behavior in real-time. This was done by using a Visual Geometry Group-16 (VGG-16) model to classify images. However, this model is computationally expensive due to the large number of parameters. The authors improved the model's performance using the Faster Region -Based CNN (Faster-RCNN), a region-based CNN architecture, and the Part Affinity Fields (PAFs) to estimate the driver's pose. This approach produced a high accuracy result (98.9% accuracy training data and 97.7% accuracy validation data) with fewer false positives. Another approach used a real-time detection system that warned the driver of their distracted behavior using GoogleNet [15]. The system initially used a variety of deep learning architectures, such as Residual

Neural Network (ResNet), VGG-16, AlexNet, and GoogleNet. After evaluating all four CNNs, GoogleNet was found to be the most effective at detecting distractions, with an accuracy of 89% and a frequency of 11 Hz.

When collecting data for a real-time system that detects distracted driving, using only one camera sometimes may lead to false alarms. To improve accuracy, [16] installed a multi-angle camera surrounding the driver. The Deep Neural Networks (DNN) was used to identify driver behavior and warn early about potential risks via mobile phone usage. The system calculates the distance between the hand and mobile device regions and issues a warning based on proximity. The system achieved an accuracy of 95.7%. In [17], the authors developed a real-time distracted driver recognizer for driver warning purposes using a simple CNN that can run on low-computation devices. This network uses fundamental deep-learning layers such as convolution, average pooling, batch normalization (BN), and global average pooling. The final layer employs the Softmax function to identify ten different driver actions with an accuracy of 99.51%. Lastly, Lin *et al.,* [18] developed a Lightweight Attention-based Network (LWANet) to achieve an optimum level between overall accuracy and computation cost. The proposed architecture uses an Inverted Residual Attention Module (IRAM) that significantly reduces computational expense. The authors have developed an Android application to transmit the TensorFlow model and evaluate the model's performance in real-time scenarios. Despite having limited trainable parameters and a small model file size, they achieved remarkable performance metrics with an accuracy of 98.45% and 99.37% on two different datasets.

It is important to note that the current studies on driver distraction are limited in their approach. They only rely on one attribute to identify distraction, which is insufficient to guarantee driver safety. To make a system more intelligent, other factors like vehicle motion, vehicle surrounding environment, vehicle speed, and distance between vehicles should also be considered. For instance, a vehicle drifting between lanes could be a sign of inattentive or intoxicated driving, and intense emotions, such as extreme anger, could also negatively affect a driver's ability to drive safely. Therefore, studies have presented other approaches [16], [17] to reduce false alarms on distracted behavior that can lead to car accidents.

2.2.3 Lightweight CNN models combined with other techniques

A deep learning algorithm that uses object detection and bi-directional feature pyramid networks (BiFPN) was discussed in *[19]* to identify drowsiness and distracted behavior in drivers. The algorithm was trained on the Driver Monitoring Dataset (DMD), which includes cognitive and manual distraction scenarios. The model employs an advanced data augmentation approach to improve its generalization ability and utilizes the original network's C3 module to enhance crucial feature information and mute irrelevant data. The neck network was enhanced with the BiFPN module to streamline multi-scale feature fusion while minimizing computational load. The resulting model demonstrates high identification accuracy, quick detection speed, and low memory usage. Lin *et al., [20]* present a methodology that includes both driving scenarios and driver behavior for a better safety-level decision. This study utilizes a 3D object identification approach, a modified MobileNetV3 algorithm, and a decision algorithm. To increase the model's accuracy and real-time performance for the MobileNetV3, the authors included a residual weighted squeeze-and-excite (RWSE) and implemented Hsigmiod activation. The decision method determines the safety level rating and considers the leading vehicle's speed and distance. Lastly, Alotaibi *et al.*, *[21]* have suggested using deep learning to solve the driver behavior detection problem. The performance of detecting driver-distracted action improved to 99.3% (StateFarm dataset) using a Hierarchical Recurrent Neural Network (HRNN) and a combined model with an Inception module and a ResNet. This technique recognizes the category of manual distractions, which include reaching behind, chatting on the phone, texting, eating, drinking, adjusting the audio, entertainment, or GPS, and doing one's hair and makeup. However, their model's components required more calculation time to analyze a single sample than ResNet and HRNN.

In summary, various techniques can be used to detect distracted drivers, such as machine learning and deep learning. Some standard machine learning algorithms include SVM, Naive Bayes, and decision trees. However, machine learning techniques exhibit reduced efficiency and accuracy compared to deep learning methods like the CNN model. Some studies have focused on increasing prediction performance using models like AlexNet, VGG-16, VGG-19, UNet, ResNet v5, and others. However, as accuracy increases, these models become unsuitable for embedded deployment due to their complexity and lack of real-time deployment architecture. Therefore, the focus should be on simple, lightweight models that embedded systems can use without significantly reducing accuracy.

3. Proposed MobileNet V2SE Fusion Model

The MobileNet V2SE model improves upon the original MobileNet V2 architecture by incorporating the SE module and an additional layer in the output layer. This helps capture channel relationships and extract complex features, improving accuracy and generalization. The model uses several techniques to enhance generalization, stability, and convergence speed, such as Global Average Pooling, dense layers with rectified linear unit (ReLU) activation, Dropout, and batch normalization. Finally, the model includes a dense output layer with softmax activation that assigns probabilities to ten specific driving behavior classes. In this section, the MobileNet V2 architecture is briefly explained. Then, the detail of MobileNet V2SE construction is discussed, which is divided into four subsections: SE module, classification layer, freeze layer, and model configuration. Lastly, the evaluation methods and simulation environment will be explained.

3.1 MobileNet V2 architecture

MobileNetV2 is a convolutional neural network architecture designed for efficient usage on mobile and edge devices. Google developed it to enhance the original MobileNetV1. The key concept behind MobileNetV2 is depthwise separable convolution, which uses an efficient building block called an Inverted Residual Block that repeats throughout the network. Each Inverted Residual Block includes several layers: depthwise convolution, batch normalization, linear bottleneck, and shortcut connection that work together to improve the model's performance.

The depthwise convolution layer applies a 3x3 kernel size and a stride of 1 to each channel in the input feature map. This layer is computationally efficient, reducing the number of parameters compared to traditional convolutional layers. After convolution, the output is normalized using batch normalization. Then, a linear bottleneck layer is added, which involves a 1x1 convolution operation, batch normalization, and linear activation. The expansion factor of the input channels determines the number of output channels in the linear bottleneck layer. The linear bottleneck layer improves the network's capacity and allows richer representations to learn.

MobileNetV2 adds a shortcut connection between the input and output of each Inverted Residual Block to maintain high accuracy while improving efficiency. This shortcut connection goes around the layers inside the block and directly adds the input to the output. Using this shortcut connection helps to propagate gradients and allows information to flow efficiently through the network. The Inverted Residual block reduces the number of parameters while maintaining high accuracy. MobileNetV2 uses an efficient layer architecture by incorporating the linear bottleneck and inverted residual structure, taking advantage of the low-rank nature of the problem.

The neural network undergoes multiple iterations of basic building blocks to obtain several final layers. These layers help convert the feature map into either class probabilities or regression outputs, depending on the task. The final layers consist of global average pooling, fully connected layers, and activation functions like softmax or sigmoid. MobileNetV2 has effectively balanced accuracy and efficiency by incorporating inverted residuals, linear bottlenecks, depthwise separable convolutions, and multi-scale feature fusion. These design choices have aided MobileNetV2 in achieving state-of-the-art performance across various tasks and benchmarks, all while operating efficiently on mobile and embedded devices. Figure 1 shows the original MobileNet V2 architecture.

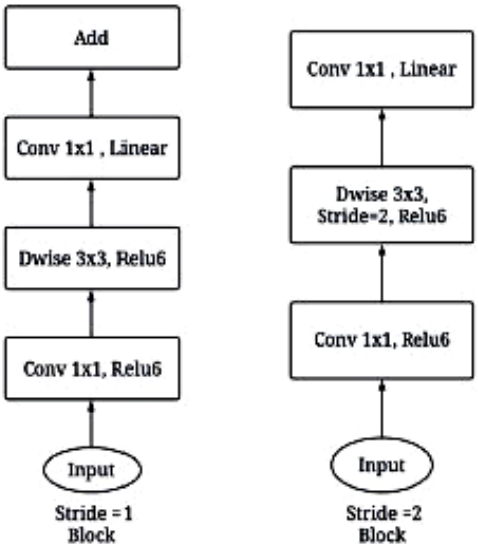

Fig. 1. MobileNetV2 architecture

3.2 MobileNet V2 architecture
3.2.1 SE module in MobileNet V2

In [22], the SE module was introduced and found to effectively improve the performance of neural networks across various tasks, including image classification, object detection, and natural language processing. This attention mechanism enhances the feature representation power of convolutional neural networks (CNNs). An advantageous aspect of the SE module is that it can be seamlessly incorporated into existing deep-learning network models without requiring extensive redesign of the network structure or parameter tuning [23].

The SE module comprises a squeezed and excited operation, as illustrated in Figure 2. The squeezed operation reduces the spatial dimensions of each feature map to a single channel by applying global average pooling. The resulting channel-wise features are then fed into an excited operation, which learns a set of channel-wise weights used to modulate each feature map's importance. The excited operation comprises two fully connected layers and a sigmoid or ReLU activation function. Element-wise multiplication applies the resulting attention scores to the feature maps.

Here, the SE modules integrate at the conclusion of inverted residual blocks, strategically placed following the depth-wise convolutional layers within each block (including Block 1, Block 2, Block 3, Block 4, and Block 5), as illustrated in Figure 3.

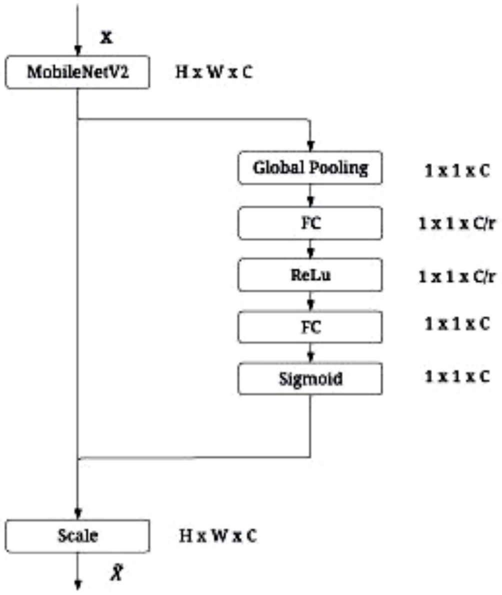

Fig. 2. A SE block

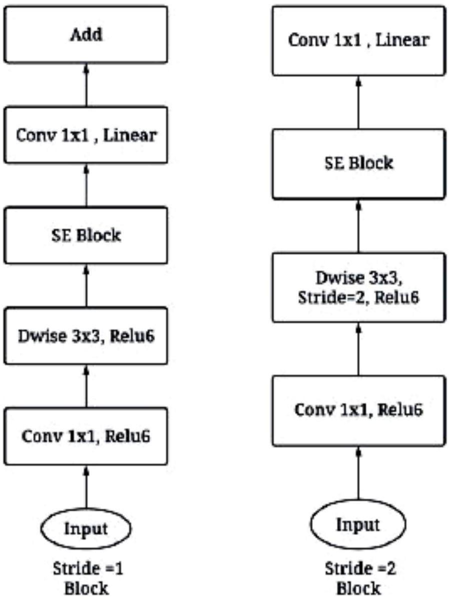

Fig. 3. MobileNetV2SE fusion

To implement the SE module, a class called SEModule is used. This class takes two input parameters: the number of channels and the reduction ratio. The reduction ratio determines how many channels will be removed in the excitation phase. The SEModule class then uses a Global Average Pooling layer to perform the squeezing operation on the input feature maps. This step reduces the dimensions of the feature maps, resulting in a channel-wise representation.

The channel-wise features are then passed through an excitation layer consisting of two fully connected (Dense) layers. The first dense layer uses ReLU activation to reduce the number of channels, while the second dense layer restores the original number of channels using sigmoid activation. This sequential excitation layer allows the model to learn channel-specific attention weights, which indicate the importance of each channel. The channel-wise attention scores are computed in the call method, which applies the squeeze layer to the input feature maps. This representation is passed through the excitation layer to obtain the channel-wise attention values. A reshape operation is performed to adjust the shape of the excitation output to match the original input dimensions. Finally, the attention scores are computed by element-wise multiplication between the input feature maps and the excitation output.

Overall, the SE module provides a modular and flexible approach to incorporating attention mechanisms within model architectures. By adapting feature maps, the SE module enables the model to focus on relevant channels and enhance its ability to recognize image patterns. However, the SE module can also increase the computational complexity of a neural network. It is crucial to evaluate the trade-offs between performance and complexity when deciding whether or not to use the SE module.

3.2.2 Classification layer V2

A Global Average Pooling layer is included to reduce the spatial dimensions of the feature maps while preserving channel-wise information. This pooling operation helps in reducing the model's complexity and parameter count, thus making it more computationally efficient. An additional dense layer with 1024 units and ReLU activation is added to extract more complex and higher-level features. This layer enhances the model's capacity to learn more sophisticated representations and differentiate between different classes.

To prevent overfitting, a dropout layer with a rate of 0.2 is applied. It randomly sets a fraction of the units to zero during training iterations. This regularization technique encourages model generalization and reduces reliance on specific features. Batch normalization is used to normalize the activations of the preceding dense layer. This enhances training stability and convergence speed by normalizing the activations. In addition, batch normalization improves the model's ability to generalize and produce reliable predictions. Finally, a dense output layer with softmax activation is incorporated to produce a probability distribution over the classes. This allows the model to assign class probabilities to input samples, facilitating classification tasks.

3.2.3 Freeze layer

When using a MobileNetV2 model, deciding which layers to freeze depends on the similarity between the current task and the task for which the model was originally trained. Freezing fewer layers can be beneficial if the target dataset significantly differs from the dataset on which the model was pre-trained. This allows more layers to be updated and enables the model to adapt to the new dataset. The best freezing configuration is determined through evaluation to identify the configuration that yields the highest accuracy and best generalization capabilities on the target dataset. In transfer learning scenarios, only a portion of the model needs further training while retaining the general features learned by the earlier layers.

3.2.4 Model configuration

The model's weights are optimized using the Adam optimizer with a learning rate of 0.0001, which is a widely used and efficient optimization algorithm. The model's performance is evaluated using the accuracy metric, and categorical cross-entropy is used as the loss function. During training, the model is trained on the training data and validated using the validation data. The number of training epochs determines how often the model iterates over the entire training dataset.

To track the model's progress and save the best-performing model weights, two callbacks are employed: checkpoint and earlystop. The checkpoint callback saves the model weights after each epoch if they yield the best validation performance so far. The earlystop callback stops the training process early if the metric does not improve over a predefined number of epochs. The class_weight parameter handles class imbalance issues in the training data by giving more importance to underrepresented classes.

By implementing these strategies, the model aims to optimize its weights using the Adam optimizer and minimize the categorical cross-entropy loss. The provided training loop ensures that the model is trained for the specified number of epochs, while the callbacks and class weights contribute to better generalization and handling of class imbalances.

3.3 Evaluation Metrics and Simulation Environment

The following metrics were used to assess MobileNet V2SE fusion performance:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

$$F1\ score = \frac{2(PPrecision \times Recall)}{Precision + Recall} \tag{4}$$

True positive (TP) refers to an instance where the model correctly predicted the positive class, True negative (TN) refers to an instance where the model correctly predicted the negative class, false positive (FP) refers to an instance where the model incorrectly predicted the positive class, and a false negative (FN) refers to an instance where the model incorrectly predicted the negative class.

$$FPS = \frac{Num\ of\ sample}{end\ time - start\ time} \tag{5}$$

Frames per second (FPS) is a commonly used metric to evaluate a model’s performance when applied to real-time data streams such as video.

The simulation was performed using a central processing unit (CPU) powered by an Intel(R) Core (TM) i5-9300H CPU @ 2.40GHz, running on the Windows 11, 64-bit operating system, and with 8GB Random Access Memory (RAM). The Python programming language was executed on an NVIDIA GeForce GTX 1050 Ti GPU. The proposed model was trained and tested on the American University in Cairo (AUC) distracted driver dataset, with 70% of the data used for training and 30% for testing purposes. Table 1 is the dataset description.

Table 1
Dataset description

Class	Description	Train	Valid	Test	No. of Image
0	Drive Safe	1708	732	266	2706
1	Texting with right-hand	913	392	133	1438
2	Using a phone with right-hand	603	259	114	976
3	Texting with left-hand	520	224	100	844
4	Using a phone with left-hand	665	285	90	1040
5	Adjust radio	527	226	90	843
6	Drink	513	220	63	796
7	Reaching behind while driving	483	208	63	754
8	Hair and Makeup	488	210	66	764
9	Conversing with passengers	965	414	138	1217

4. Results and Discussion

Table 2 shows the results of the classification model used to identify distracted driver behavior. These tables present information about precision, recall values, and F1 scores. Regarding precision, most classes had a low rate of false positives, except for C0, C6, C8, and C9. The recall values indicate that most classes successfully identified positive instances, although C0, C6, C8, and C9 had more false negatives. F1 scores were high across most classes, indicating strong overall performance in terms of both precision and recall. The different levels of difficulty of the classification tasks can explain the discrepancies observed in the F1-scores between classes C1 to C4 and classes C6 to C9. Classifying distracted driver behavior presents challenges due to the wide range of behaviors and their varying levels of subtlety.

Table 3 presents a comparison of MobileNet V2 and MobileNet V2SE models using macro average (M.A) and weighted average (W.A) values.

Table 2
Precision, recall and fi score for MobileNet v2 and MobileNet V2se

Class	MobileNet V2			MobileNet V2SE		
	Precision	Recall	F1	Precision	Recall	F1
0	0.60	0.6	0.67	0.98	0.86	0.91
1	0.85	0.85	0.90	0.93	1.00	0.96
2	0.66	0.66	0.79	0.98	1.00	0.99
3	0.60	0.60	0.74	0.98	1.00	0.99
4	1.00	1.00	0.81	0.94	1.00	0.97
5	0.91	0.91	0.94	1.00	1.00	1.00
6	0.75	0.75	0.71	0.79	0.89	0.84
7	0.94	0.94	0.48	0.77	0.98	0.86
8	0.89	0.89	0.74	0.98	0.76	0.85
9	0.46	0.46	0.55	0.86	0.87	0.86

Table 3
Macro average and weighted average for MobileNet V2 and MobileNet V2SE

Metrics	MobileNet V2		MobileNet V2SE	
	M.A	W.A	M.A	W.A
Precision	0.76	0.79	0.82	0.93
Recall	0.77	0.72	0.94	0.93
F1 Score	0.73	0.73	0.92	0.93

The confusion matrix, shown in Figure 4, indicates that the model faced difficulties in accurately identifying "drinking" and "makeup/hair" instances. Specifically, the model misclassified seven instances of "drinking" as other behaviors, and 16 instances of "makeup" application were misclassified as different behaviors. These findings highlight the model's relatively lower accuracy in detecting these two specific types of behavior.

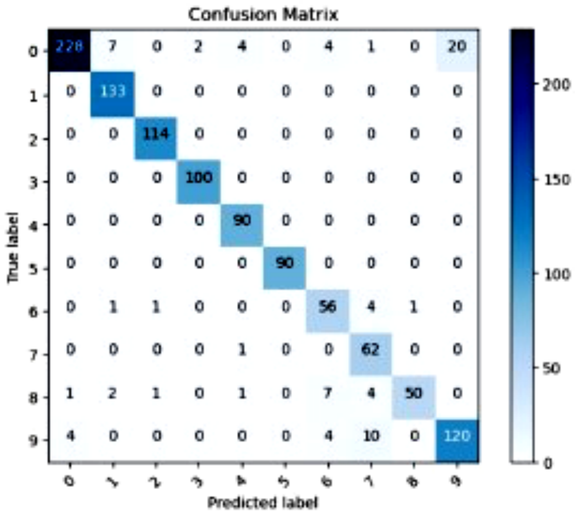

Fig. 4. Confusion matrix of MobileNet V2SE

Figure 5 shows that the training loss curve decreases and stabilizes over time. Similarly, the validation loss curve also decreases and reaches a stable point, with only a small difference compared to the training loss curve. This indicates that the model is performing well. It is well-balanced between complexity and generalization, and no sign of overfitting or underfitting, since both training and validation performances remain reasonable and consistent. Overall, the results demonstrate a harmonious relationship between training and validation performance, the model is effective.

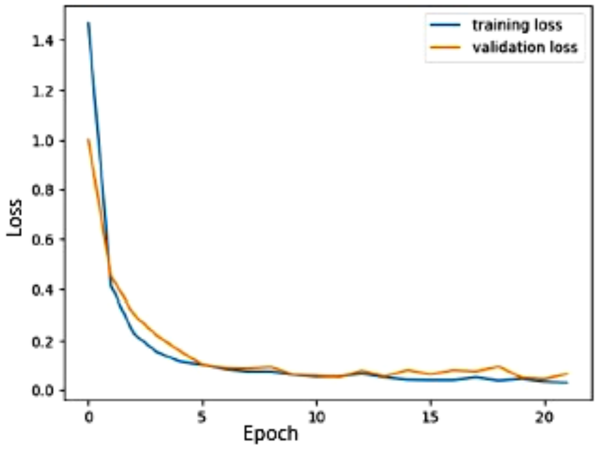

Fig. 5. The loss of MobileNet V2SE during training and validation

The MobileNet V2SE achieved an FPS measurement of 53, while the original MobileNetV2 achieved 55.64 FPS when run on an NVIDIA GeForce GTX 1050 GPU, as shown in Table 4. The difference can be attributed to the addition of the SE module in the MobileNetV2. The SE module introduces extra learnable weights and biases associated with the fully connected layers in the excitation operation, which boosts the model's performance on various tasks. However, the increase in parameters count results in additional computation during both training and inference phases, leading to more calculations and a higher computational workload.

Table 4
FRS and parameters

Model	FRS	Total parameters
MobileNet V2	55.65	2,257,984
MoileNet V2SE	53.00	3,790,234

The results presented in Table 5 show that the MobileNet V2SE approach achieves an accuracy of 93%, which is higher than MobileNetV2 in [24] with 89.38% and the initial finetuning phase with only 72%. This means that the "SE" module enhances the overall performance of the MobileNetV2 model. Furthermore, the MobileNet V2SE performs better than pre-trained models like AlexNet [13], InceptionV3 [13], and VGG [25] in terms of balancing accuracy and parameter utilization. The MobileNet V2SE achieves a desirable balance between accuracy and model complexity, highlighting the potential for size reduction while preserving competitive accuracy. However, when compared to other modified models like modified VGG [25], MobileVGG [26], LWANet [18], Lightweight CNN [27], MobileNetV2-tiny [28], and EFFNet [29], MobileNet V2SE exhibits slightly lower performance. Nevertheless, the MobileNet V2SE demonstrates proficiency with specific modified models such as InceptionV3 + BiLSTM [10] and Inception + HRNN [21].

According to Table 5, MobileNet V2SE performs moderately when compared to other modified models. Different models have varying ways of performing due to differences in their structures, such as the number of layers, receptive fields, and feature extraction capabilities. These differences directly affect a model's accuracy. Additionally, a model's ability to understand complex patterns in data depends on the number of parameters it has. Models with more parameters, such as VGG, can identify intricate features better, but they are more prone to overfitting when trained with limited data. The "SE" module in MobileNetV2 can enhance a model's performance. These modules allow the model to prioritize and focus on more informative features, which can improve its accuracy.

Table 5
Comparison of MobileNet V2SE with others models

Model	Accuracy (%)	Parameters (million)	Dataset
AlexNet [13]	93.65	62	AUC
Inception [13]	95.17	24	AUC
VGG [25]	94.44	140	AUC
Modified VGG [25]	95.54	15	AUC
Inception V3 + BiLSTM [10]	92.77	24.33	AUC
LWANet [18]	98.45	1.22	AUC
Lightweight CNN [27]	95.36	0.46	AUC
Inception + HRNN [21]	92.36	-	AUC
MobileVGG [26]	95.24	2.20	AUC
MobileNet V2-tiny [28]	94.77	2.78	AUC
EFFNet [29]	98.97	4.57	AUC2
MobileNet V2 [24]	89.38	3.50	AUC
MobileNet V2 (fined-tunned)	72.00	3.50	AUC
MobileNet V2SE	93.00	3.75	AUC

5. Conclusion

Distracted driving leads to many injuries and deaths each year. To address this issue, a model called the MobileNetV2 network has been introduced to detect distracted drivers. This study aimed to improve the accuracy of the lightweight MobileNetV2 model. This was achieved by analysing the MobileNetV2 model and its hyperparameters and finding values that would improve accuracy. The model's performance was evaluated using different configurations of hyperparameters, leading to the identification of the best values to maximize accuracy. The best settings for batch size, learning rate, number of epochs, and optimizer were 32, 0.0001, 21, and Adam, respectively.

SE modules were added to the MobileNetV2 model to improve its accuracy. This was done by placing the SE module after the depth-wise convolutional layers in each block, resulting in a significant accuracy improvement of 21%. The MobileNet V2SE model was compared to other approaches to analyse its accuracy. It was found that the MobileNet V2SE model had a balance between accuracy and parameter utilization, outperforming pre-trained models like AlexNet, InceptionV3, and VGG. It achieved competitive accuracy with fewer parameters compared to VGG and Inception V3. However, the MobileNet V2SE model showed slightly lower performance than other modified models.

The MobiltNet V2SE has achieved high accuracy, but may not work as well in low-lighting conditions. This is because the dataset used to train the model was captured in brighter settings. To address this, it would be helpful to collect a different dataset that covers driver behavior during both daytime and nighttime, especially in dimly lit environments. In the future, this technology can be used in different real-world scenarios, including computer vision systems to monitor and analyze driver conditions in real time. By adding driver conditions as an extra feature in car safety systems, it can improve their effectiveness and responsiveness, making driving safer for everyone.

Acknowledgement

The authors would like to acknowledge the Faculty of Engineering, Universiti Malaysia Sarawak, for the support in the preparation of this article. This research was not funded by any grant.

References

[1] WHO, "Road traffic injuries," World Health Organization, 20 June 2022. [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/road-traffic-injuries. [Accessed 3 October 2023].

[2] NHTSA, "Distracted Driving Dangers and Statistics," United States Department of Transportation, 2020. [Online]. Available: https://www.nhtsa.gov/risky-driving/distracted-driving. [Accessed 5 October 2023].

[3] Yellman, Merissa A., Leah Bryan, Erin K. Sauber-Schatz, and Nancy Brener. "Transportation risk behaviors among high school students-Youth risk behavior survey, United States, 2019." MMWR supplements 69, no. 1 (2020): 77. https://doi.org/10.15585/mmwr.su6901a9

[4] Kashevnik, Alexey, Roman Shchedrin, Christian Kaiser, and Alexander Stocker. "Driver distraction detection methods: A literature review and framework." IEEE Access 9 (2021): 60063-60076. https://doi.org/10.1109/ACCESS.2021.3073599

[5] Feng, Demeng, and Yumeng Yue. "Machine Learning Techniques for Distracted Driver Detection." (2019).

[6] Torres, Renato, Orlando Ohashi, and Gustavo Pessin. "A machine-learning approach to distinguish passengers and drivers reading while driving." Sensors 19, no. 14 (2019): 3174. https://doi.org/10.3390/s19143174

[7] Ezzouhri, Amal, Zakaria Charouh, Mounir Ghogho, and Zouhair Guennoun. "Robust deep learning-based driver distraction detection and classification." IEEE Access 9 (2021): 168080-168092. https://doi.org/10.1109/ACCESS.2021.3133797

[8] Bahari, Muhammad Saiful Haqem Saiful, and Lucyantie Mazalan. "Distracted driver detection using deep learning." In 2022 IEEE 18th International Colloquium on Signal Processing & Applications (CSPA), pp. 198-203. IEEE, 2022. https://doi.org/10.1109/CSPA55076.2022.9781938

[9] Abbas, Tahir, Syed Farooq Ali, Mazin Abed Mohammed, Aadil Zia Khan, Mazhar Javed Awan, Arnab Majumdar, and Orawit Thinnukool. "Deep learning approach based on residual neural network and SVM classifier for driver's distraction detection." Applied Sciences 12, no. 13 (2022): 6626. https://doi.org/10.3390/app12136626

[10] Mase, Jimiama Mafeni, Peter Chapman, Grazziela P. Figueredo, and Mercedes Torres Torres. "A hybrid deep learning approach for driver distraction detection." In 2020 International Conference on Information and Communication Technology Convergence (ICTC), pp. 1-6. IEEE, 2020. https://doi.org/10.1109/ICTC49870.2020.9289588

[11] Sajid, Faiqa, Abdul Rehman Javed, Asma Basharat, Natalia Kryvinska, Adil Afzal, and Muhammad Rizwan. "An efficient deep learning framework for distracted driver detection." IEEE Access 9 (2021): 169270-169280. https://doi.org/10.1109/ACCESS.2021.3138137

[12] Ghoddoosian, Reza, Marnim Galib, and Vassilis Athitsos. "A realistic dataset and baseline temporal model for early drowsiness detection." In Proceedings of the ieee/cvf conference on computer vision and pattern recognition workshops, pp. 0-0. 2019. https://doi.org/10.1109/CVPRW.2019.00027

[13] Abouelnaga, Yehya, Hesham M. Eraqi, and Mohamed N. Moustafa. "Real-time distracted driver posture classification." arXiv preprint arXiv:1706.09498 (2017).

[14] Draz, Hafiz Umer, Muhammad Zeeshan Khan, Muhammad Usman Ghani Khan, Amjad Rehman, and Ibrahim Abunadi. "A Novel Ensemble Learning Approach of Deep Learning Techniques to Monitor Distracted Driver Behaviour in Real Time." In 2021 1st International Conference on Artificial Intelligence and Data Analytics (CAIDA), pp. 251-256. IEEE, 2021. https://doi.org/10.1109/CAIDA51941.2021.9425243

[15] Tran, Duy, Ha Manh Do, Weihua Sheng, He Bai, and Girish Chowdhary. "Real-time detection of distracted driving based on deep learning." IET Intelligent Transport Systems 12, no. 10 (2018): 1210-1219. https://doi.org/10.1049/iet-its.2018.5172

[16] Jin, Chongchong, Zhongjie Zhu, Yongqiang Bai, Gangyi Jiang, and Anqing He. "A deep-learning-based scheme for detecting driver cell-phone use." IEEE access 8 (2020): 18580-18589. https://doi.org/10.1109/ACCESS.2020.2968464

[17] Nguyen, Duy-Linh, Muhamad Dwisnanto Putro, and Kang-Hyun Jo. "Distracted driver recognizer with simple and efficient convolutional neural network for real-time system." In 2021 21st International Conference on Control, Automation and Systems (ICCAS), pp. 371-375. IEEE, 2021. https://doi.org/10.23919/ICCAS52745.2021.9649760

[18] Lin, Yingcheng, Dingxin Cao, Zanhao Fu, Yanmei Huang, and Yanyi Song. "A Lightweight Attention-Based Network towards Distracted Driving Behavior Recognition." Applied Sciences 12, no. 9 (2022): 4191. https://doi.org/10.3390/app12094191

[19] T. Li, Y. Zhang, Q. Li, and T. Zhang, "AB-DLM: An Improved Deep Learning Model Based on Attention Mechanism and BiFPN for Driver Distraction Behavior Detection," IEEE Access, vol. 10, p. 83138-83151, 2022. https://doi.org/10.1109/ACCESS.2022.3197146

[20] Lin, Peng-Wei, and Chih-Ming Hsu. "Innovative Framework for Distracted-Driving Alert System Based on Deep Learning." IEEE Access 10 (2022): 77523-77536. https://doi.org/10.1109/ACCESS.2022.3186674

[21] Alotaibi, Munif, and Bandar Alotaibi. "Distracted driver classification using deep learning." Signal, Image and Video Processing 14, no. 3 (2020): 617-624. https://doi.org/10.1007/s11760-019-01589-z

[22] Hu, Jie, Li Shen, and Gang Sun. "Squeeze-and-excitation networks." In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 7132-7141. 2018. https://doi.org/10.1109/CVPR.2018.00745

[23] Zhu, Xizhou, Dazhi Cheng, Zheng Zhang, Stephen Lin, and Jifeng Dai. "An empirical study of spatial attention mechanisms in deep networks." In Proceedings of the IEEE/CVF international conference on computer vision, pp. 6688-6697. 2019. https://doi.org/10.1109/ICCV.2019.00679

[24] Hossain, Md Uzzol, Md Ataur Rahman, Md Manowarul Islam, Arnisha Akhter, Md Ashraf Uddin, and Bikash Kumar Paul. "Automatic driver distraction detection using deep convolutional neural networks." Intelligent Systems with Applications 14 (2022): 200075. https://doi.org/10.1016/j.iswa.2022.200075

[25] Baheti, Bhakti, Suhas Gajre, and Sanjay Talbar. "Detection of distracted driver using convolutional neural network." In Proceedings of the IEEE conference on computer vision and pattern recognition workshops, pp. 1032-1038. 2018. https://doi.org/10.1109/CVPRW.2018.00150

[26] Baheti, Bhakti, Sanjay Talbar, and Suhas Gajre. "Towards computationally efficient and realtime distracted driver detection with mobilevgg network." IEEE Transactions on Intelligent Vehicles 5, no. 4 (2020): 565-574. https://doi.org/10.1109/TIV.2020.2995555

[27] Nguyen, Duy-Linh, Muhamad Dwisnanto Putro, Xuan-Thuy Vo, and Kang-Hyun Jo. "Light-weight convolutional neural network for distracted driver classification." In IECON 2021-47th Annual Conference of the IEEE Industrial Electronics Society, pp. 1-6. IEEE, 2021. https://doi.org/10.1109/IECON48115.2021.9589212

[28] Zhuang, Qinghe, Zhehao Dai, and Jia Wu. "Deep active learning framework for lymph node metastasis prediction in medical support system." Computational Intelligence and Neuroscience 2022 (2022). https://doi.org/10.1155/2022/4601696

[29] Khan, Taimoor, Gyuho Choi, and Sokjoon Lee. "EFFNet-CA: an efficient driver distraction detection based on multiscale features extractions and channel attention mechanism." Sensors 23, no. 8 (2023): 3835. https://doi.org/10.3390/s23083835

[30] Razak, Siti Fatimah Abdul, Sumendra Yogarayan, Azlan Abdul Aziz, Mohd Fikri Azli Abdullah, and Noor Hisham Kamis. "Physiological-based Driver Monitoring Systems: A Scoping Review." Civil Engineering Journal 8, no. 12 (2022): 3952-3967. https://doi.org/10.28991/CEJ-2022-08-12-020