Deep learning based driver distraction detection

Kathar Patcha Abdul Rahim
School of Computer Science
University of Windsor
Windsor, Canada
abdulrak@uwindsor.ca

Manohar Reddy Tippireddy
School of Computer Science
University of Windsor
Windsor, Canada
tippirem@uwindsor.ca

Asif Kasim Lohar
School of Computer Science
University of Windsor
Windsor, Canada
lohara@uwindsor.ca

Dr. Alioune Ngom
School of Computer Science
University of Windsor
Windsor, Canada
angom@uwindsor.ca

***Abstract*— In recent years, the increasing demands of daily life have made it challenging for individuals to stay consistently alert and focused, particularly while driving. This lack of attention can lead to various mistakes on the road, such as drowsiness, falling asleep, or being distracted by a phone call. Such situations significantly increase the risk of road accidents, injuries, and fatalities, which are more common than often realized. To address these critical issues, it is imperative to develop effective solutions that enhance driving safety by understanding driver decisions and behaviors. This study proposes a Driver Activity Recognition (DAR) system designed to identify and classify various driver activities using a deep learning approach. The system employs Convolutional Neural Networks (CNN) to distinguish between ten common driving activities. These activities include normal driving and several distracted behaviors such as talking on the phone, texting, eating or drinking, adjusting the radio, touching the face, applying makeup, turning around, and engaging in conversation with passengers. By accurately recognizing these behaviors, the DAR system aims to reduce the likelihood of accidents caused by distracted driving and improve overall road safety.**

Keywords— Driver Behavior; Convolutional Neural Network; Deep Learning, Driver Distraction.

I. INTRODUCTION

In driving safety, the driver's behavior plays a significant role, leading to serious outcomes. To decrease the number of road accidents and to better driving safety, an architecture for real-time Driver Activity Recognition is proposed in this study. According to a survey, it is stated that around 89% of the times when accidents occur the root reason is nothing but unethical behavior of human drivers, and it is also stated that we can significantly decrease this percentage with the help of an accurate monitoring system to detect the behavior of drivers. Thus, distracted DAR became one of the most important tasks in recent times. Regarding the highly automated vehicle where the driver is given full controls over the vehicle in certain emergency conditions, this real-time driver behavior tracking would help in deciding whether the driver can takeover or not in such scenarios. To overcome such problems, a deep learning-based DAR system is proposed in this study, to understand the driver behaviors according to the requirements.

The main objective of this project is to prepare a model which can detect the driver's behavior with utmost accuracy and helps in monitoring the driver's behavior continuously. So that it also can be used in various real-world applications. It also helps in various situations by suggesting better decisions for drivers to have a better journey indirectly resulting in fewer accidents or injuries.

II. LITERATURE REVIEW

1. Deep Learning and CNNs

Convolutional Neural Networks (CNNs) have revolutionized image processing and classification tasks, making them highly suitable for driver distraction detection. Shamsaldin, Fattah, Rashid, and Al-Salihi discussed the diverse applications of CNNs, including scene labeling, face recognition, and image classification, which are foundational to understanding and implementing driver distraction detection systems [6].

Gupta, Pathak, and Kumar provided a comprehensive review of deep learning and transfer learning, highlighting the superior performance of deep learning systems in image processing and pattern recognition tasks [7]. Similarly, Shetty, Varma, Navi, and Ahmed traced the history, evolution, and applications of deep learning, emphasizing its role in computer vision and natural language processing [8].

2. Driver Distraction Detection Techniques

Hossain, Rahman, and Islam utilized deep convolutional neural networks for automatic driver distraction detection, highlighting the effectiveness of CNNs in identifying various distracted driving behaviors [1]. Another study by Chawan and Satardekar focused on distracted driver detection and classification, reinforcing the potential of deep learning models in this domain [2].

Xing, Lv, Wang, Cao, Velenis, and Wang explored driver activity recognition for intelligent vehicles using a deep learning approach, which is crucial for developing systems that can recognize and respond to driver behaviors in real-time [4]. Furthermore, Abouelnaga, Eraqi, and Moustafa introduced the AUC Distracted Driver Dataset, providing a valuable resource for training and evaluating deep learning models in driver distraction detection [5].

3. Datasets and Model Performance

The State Farm Distracted Driver Detection dataset has been extensively used in research to train and test various models [3]. This dataset includes images labeled with different driving activities, which are essential for developing robust driver distraction detection systems.

Qi, Larochelle, Huet, Luo, and Yu discussed the advancements in deep learning for multimedia computing, highlighting the development of deep networks that capture dependencies between different data genres [9]. Aishwarya and Kumar provided insights into the applications of deep learning across various domains, emphasizing its significance in pattern recognition and image processing [10].

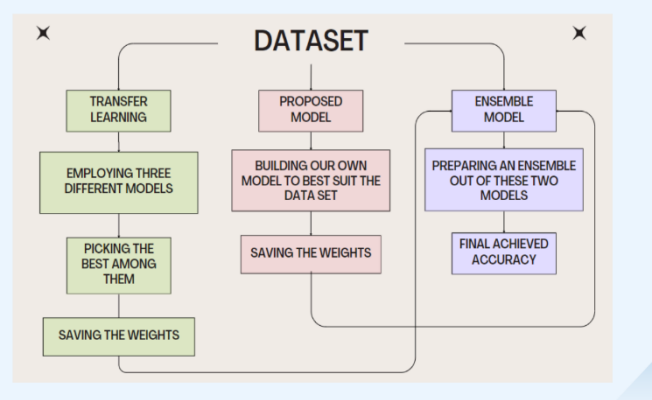
Vargas, Mosavi, and Ruiz reviewed the hierarchical structure of deep learning, which allows for high-level feature extraction from low-level features, making it a powerful tool for driver distraction detection [11]. Phalke and Jahirabadkar surveyed deep learning techniques for Near Duplicate Video Retrieval, illustrating the broad applicability of CNNs in various data-intensive tasks [12].

4. Applications in Healthcare and Other Domains

Shelke, Kumar, Karetla, Sulthana, Beohar, and Pant explored the application of deep learning techniques in enhancing patient treatment facilities in healthcare, demonstrating the versatility and impact of these algorithms beyond driver distraction detection [13].

III. Methodology

In this study, we propose a comprehensive approach to driver activity recognition using advanced deep learning techniques. The dataset is the cornerstone of our methodology, from which we derive various models and strategies to achieve optimal performance.



1. Data Collection

The State Farm Distracted Driver Detection dataset is used for this project. This dataset contains 22,500 labeled RGB images with a resolution of 640x480 pixels. The dataset comprises ten classes: normal driving, texting with the right hand, calling with the right hand, texting with the left hand, calling with the left hand, operating the radio, drinking, reaching behind, hair and makeup, and talking to passengers [3]. These images are crucial for training the model to recognize diverse types of driver behaviors.

2. Data Preprocessing

Data preprocessing is a critical step to ensure the quality and consistency of the dataset. The preprocessing steps include:

a) **Image Resizing**: All images are resized to a uniform size suitable for the deep learning model input.

b) **Normalization**: Pixel values are normalized to a range of 0 to 1 to facilitate faster convergence during training.

c) **Data Augmentation**: Techniques such as rotation, translation, and flipping are applied to increase the variability of the training data and reduce overfitting.

3. Transfer Learning and Model Selection

This study employs transfer learning using three pre-trained models: DenseNet121, VGG16, and MobileNetV2. These models are selected due to their proven performance in image classification tasks [5]. The following steps outline the training process:

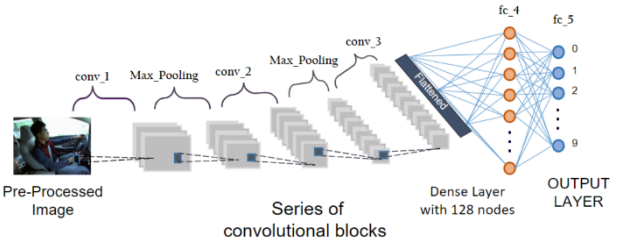
a) **Model Initialization**: The pre-trained models are initialized with weights from training on the ImageNet dataset.

b) **Fine-Tuning**: The models are fine-tuned on the distracted driving dataset by adjusting their weights to better suit the specific task.

4. Custom CNN Architecture

In addition to the pre-trained models, a custom CNN architecture named DARNET is developed. This model starts with a convolutional layer with 64 filters followed by MaxPooling2D layers. Further convolutional layers with 32 filters continue feature extraction. The network ends with a Flatten layer and Dense layers for classification [6].

Training Parameters: The models are trained with a learning rate scheduler to optimize convergence. The number of epochs and batch size are determined through experimentation to balance training time and model performance.



The diagram above illustrates the architecture of the Convolutional Neural Network (CNN) utilized for the Driver Activity Recognition (DAR) system. This architecture is designed to process and classify images of drivers into different activity categories. The following components are depicted in the image:

a. Input Layer (Pre-Processed Image):

The initial stage involves feeding a pre-processed image into the model. The pre-processing steps typically include resizing the image to a standard size, normalizing pixel values, and augmenting the data to enhance model robustness.

b. Convolutional Blocks:

The input image is passed through a series of convolutional blocks, each comprising a convolutional layer followed by an activation function (usually ReLU) and a max-pooling layer.

Conv_1 and Max_Pooling: The first convolutional layer (conv_1) extracts basic features such as edges and textures from the image. This layer is followed by a max-pooling operation that reduces the spatial dimensions, retaining the most significant information.

Conv_2 and Max_Pooling: The second convolutional layer (conv_2) captures more complex patterns and shapes. Another max-pooling layer follows this to further down-sample the feature maps.

Conv_3 and Max_Pooling: The third convolutional layer (conv_3) captures even higher-level features. The subsequent max-pooling layer further compresses the data, facilitating more abstract feature extraction.

c. Flattening:

The feature maps resulting from the final convolutional block are flattened into a one-dimensional vector. This transformation is essential for connecting the convolutional blocks with the fully connected (dense) layers.

d. Fully Connected (Dense) Layers:

The flattened vector is passed through a dense layer with 128 nodes. This layer is fully connected, meaning each node is connected to every output from the previous layer, enabling the model to learn complex representations of the data.

Two more fully connected layers, denoted as fc_4 and fc_5, follow this dense layer. These layers continue to process and refine the features extracted by the convolutional layers.

e. Output Layer:

The final output layer consists of 10 nodes, corresponding to the 10 different driver activities being classified. Each node represents a specific class, such as normal driving or various forms of distracted driving. The output layer uses a softmax activation function to produce a probability distribution over the classes, indicating the model's confidence in each prediction.

This CNN architecture efficiently extracts and classifies driver activities from images, leveraging multiple layers of feature extraction and abstraction. The combination of convolutional layers and dense layers allows the model to learn and distinguish between subtle differences in driver behaviors, contributing to a robust driver activity recognition system.

5. Ensemble Learning

Ensemble learning is employed to enhance the performance of the models. The predictions from DenseNet121 and the custom CNN (DARNET) are combined using an average ensembling technique. This method leverages the complementary strengths of both models, improving overall accuracy and robustness [7].

6. Performance Evaluation

The performance of the models is evaluated using several metrics:

a) **Accuracy**: The overall accuracy of each model is calculated.

b) **Confusion Matrix**: Confusion matrices are generated to analyze the classification results in detail, highlighting true positives, false positives, true negatives, and false negatives.

c) **Loss and Accuracy Curves**: Training and validation loss and accuracy curves are plotted to visualize the model's performance over epochs.

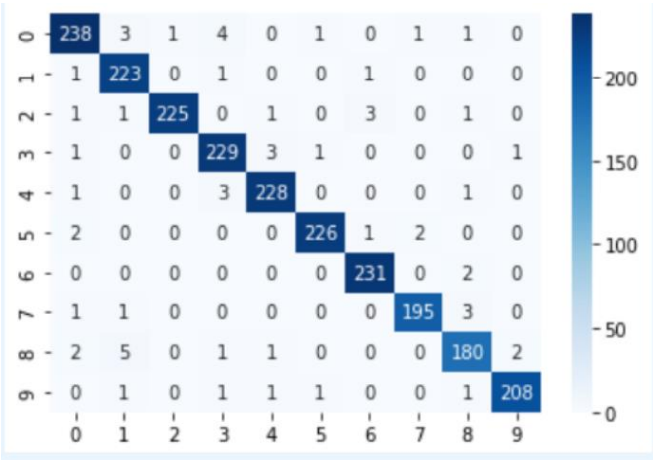
IV. Results

1. Performance of Pre-trained Models

A. DenseNet121

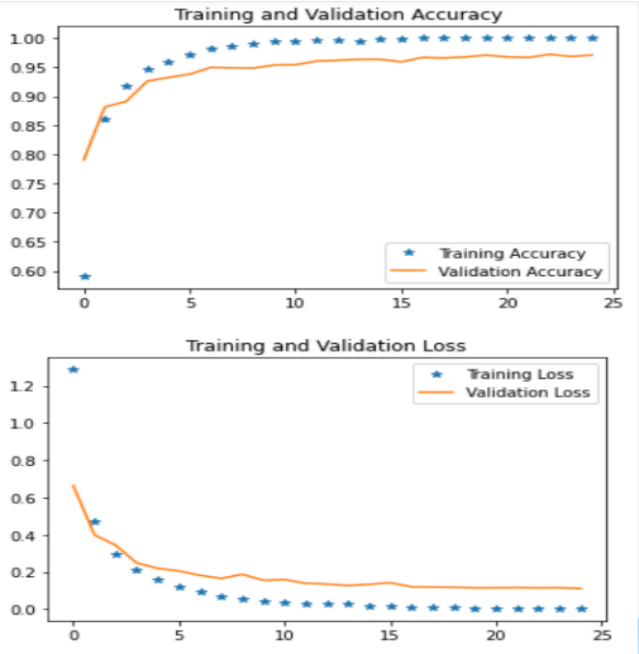
DenseNet121, known for its deep connectivity pattern, was fine-tuned using the State Farm dataset. The model demonstrated robust performance in recognizing distracted driving behaviors, achieving high accuracy rates. The confusion matrix for DenseNet121 indicated strong classification ability across most categories, with minimal false positives and negatives. This model's deep architecture allowed it to leverage the complex spatial hierarchies in the image data, contributing to its superior performance in identifying specific driver behaviors such as texting, calling, and operating the radio.

The learning rate was initially set at a moderate level to ensure a balance between rapid convergence and the prevention of overfitting. Throughout the training process, adjustments were made to optimize performance, leading to improved accuracy metrics. The loss and accuracy curves plotted during training and validation phases showed consistent improvement, with DenseNet121 maintaining high accuracy and low loss rates across multiple epochs.



The confusion matrix illustrates the performance of DenseNet121 in classifying driver activities. Each row of the matrix represents the true class, while each column represents the predicted class. The diagonal elements show the number of correct predictions for each class. High values along the diagonal indicate accurate classifications, while off-diagonal elements represent misclassifications. The matrix shows that

DenseNet121 has high classification accuracy, with most values concentrated along the diagonal, indicating the model's robustness in identifying various driver activities.



The training and validation accuracy graph shows the accuracy of the DenseNet121 model over each epoch during training. The x-axis represents the number of epochs, while the y-axis shows the accuracy percentage. The blue stars indicate training accuracy, and the orange line indicates validation accuracy. The graph demonstrates a steady increase in both training and validation accuracy, with the model achieving high accuracy levels towards the end of the training process. This indicates that the model generalizes well to unseen data and effectively learns the patterns necessary for accurate classification.

The training and validation loss graph shows the loss values of the DenseNet121 model over each epoch during training. The x-axis represents the number of epochs, and the y-axis shows the loss values. The blue stars represent training loss, and the orange line represents validation loss. The graph shows a consistent decrease in both training and validation loss, indicating that the model is learning effectively and minimizing errors over time. The low final loss values suggest that the model has achieved a good fit to the training data and is likely to perform well on new, unseen data.

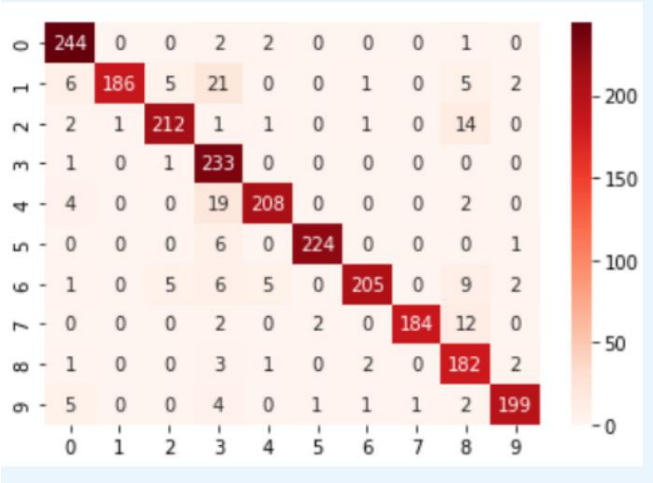
```
# Getting the accuracy of DenseNet model.
test_loss, test_acc = Dense_net.evaluate(test_images, test_labels)
print("Accuracy of the densenet model:",test_acc*100,"%")
```

```
71/71 [==============================] - 6s 62ms/step - loss: 0.0950 - ac
Accuracy of the densenet model: 97.2804307937622 %
```

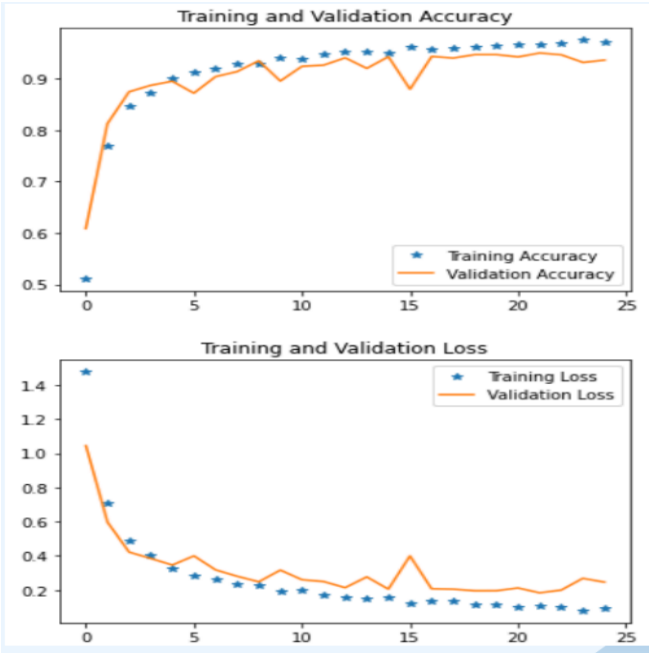
The output section displays the final accuracy of the DenseNet121 model after evaluation on the test dataset. The test accuracy is shown to be 97.28%, indicating that the model performs exceptionally well in classifying driver activities. The low-test loss of 0.0950 further supports the model's robustness and effectiveness in accurately predicting various driver behaviors, making it a reliable tool for driver distraction detection.

B. VGG16

VGG16, characterized by its simplicity and depth, also performed well in the task of detecting distracted driving behaviours. Although slightly less accurate than DenseNet121, VGG16 showed consistent results across dissimilar categories. Its architecture, consisting of 16 convolutional layers, enabled it to capture essential features necessary for distinguishing between normal and distracted driving activities. The confusion matrix for VGG16 highlighted a balanced performance, with higher misclassification rates in some categories compared to DenseNet121. However, the overall accuracy remained competitive, demonstrating VGG16's efficacy in real-time driver behavior monitoring systems.



The confusion matrix shows the performance of the VGG16 model on a classification task. It accurately classifies most instances, with high values along the diagonal, indicating correct predictions. Misclassifications are minimal, with some off-diagonal values showing errors between certain classes. For instance, class 0 has 244 correct predictions, class 1 has 186, and so on, with only a few misclassifications per class.



The training and validation accuracy graph shows

that both accuracies increase rapidly in the initial epochs and start to plateau around epoch 10. The training accuracy reaches close to 100%, while the validation accuracy stabilizes around 90%. This indicates good model performance with slight overfitting as the training accuracy is higher than validation accuracy.

The training and validation loss graph indicates a rapid decrease in both losses during the initial epochs, stabilizing around epoch 10. The training loss drops to 0, while the validation loss converges slightly above the training loss, indicating the model's convergence with minimal overfitting.

```
# Getting the accuracy of VGG16 model.
test_loss, test_acc = vgg_net.evaluate(test_images, test_labels)
print("Accuracy of the vgg16 model:",test_acc*100,"%")
```

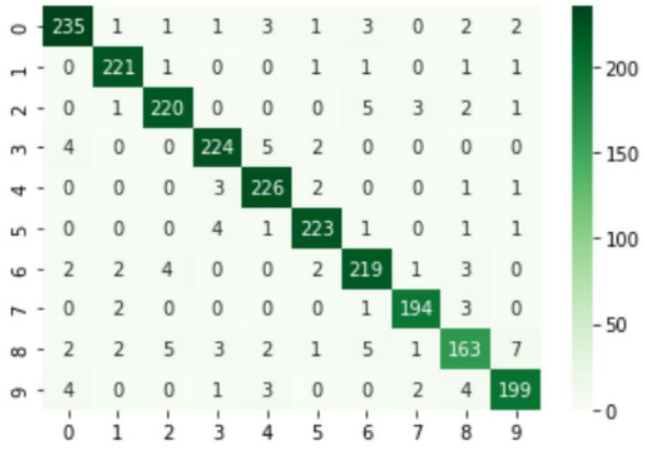
```
71/71 [==============================] - 2s 26ms/step - loss: 0.2703 - ac
Accuracy of the vgg16 model: 92.5991952419281 %
```

The final test accuracy of the VGG16 model is 92.60%, with a loss of 0.2703. This high accuracy indicates that the model performs well on the test set, validating its effectiveness in the classification task.

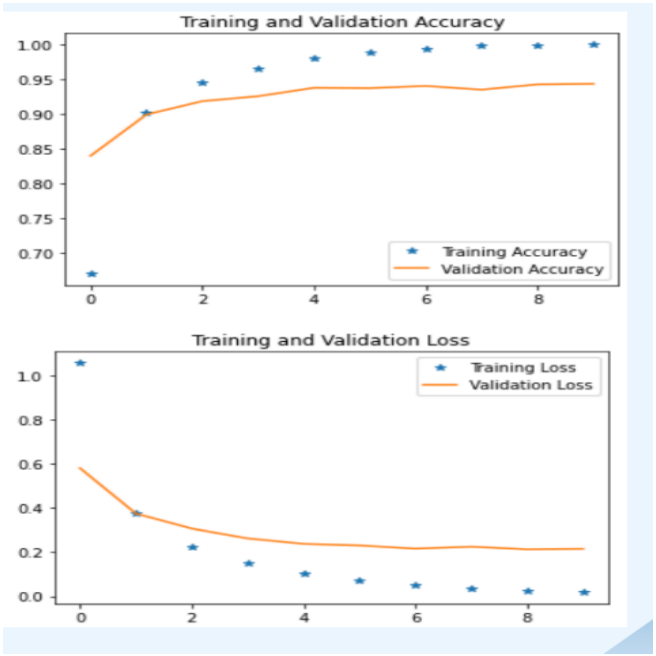
C.

D. MobileNetV2

MobileNetV2, designed for mobile and embedded vision applications, provided a good balance between accuracy and computational efficiency. The model's architecture, which includes depthwise separable convolutions, allowed it to perform well with fewer parameters, making it suitable for real-time applications where computational resources are limited. While its performance was competitive, MobileNetV2 did not surpass DenseNet121 and VGG16. The confusion matrix for MobileNetV2 showed strengths in certain categories but also highlighted areas where it struggled compared to the other models. The accuracy and loss curves indicated stable training with consistent improvements across epochs.



The confusion matrix for the MobileNetV2 model shows high accuracy with most values concentrated along the diagonal, indicating correct predictions. For example, class 0 has 235 correct predictions, class 1 has 221, and class 2 has 220, with minimal misclassifications in other classes.



The training and validation accuracy graph shows both accuracies increasing rapidly in the initial epochs and stabilizing around epoch 5. The training accuracy approaches 100%, while the validation accuracy remains close to 95%, indicating strong model performance with minimal overfitting.

The training and validation loss graph demonstrates a rapid decrease in both losses during the initial epochs, leveling off around epoch 5. The training loss approaches 0, while the validation loss remains slightly higher but stable, suggesting good model convergence.

```
# Getting the accuracy of resnet model.
test_loss, test_acc = mb_net.evaluate(test_images, test_labels)
print("Accuracy of the MobileNetV2 model:",test_acc*100,"%")
```

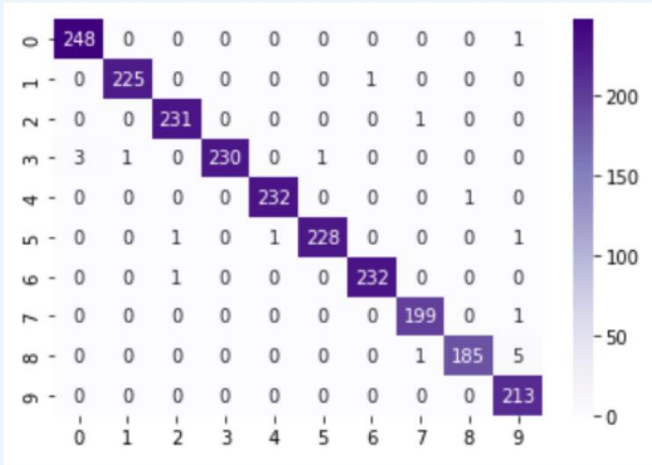
```
71/71 [==============================] - 2s 24ms/step - loss: 0.1860 -
Accuracy of the MobileNetV2 model: 94.69460248947144 %
```

The final test accuracy of the MobileNetV2 model is 94.69%, with a loss of 0.1860. This high accuracy indicates that the model performs exceptionally well on the test set, confirming its robustness in the classification task.

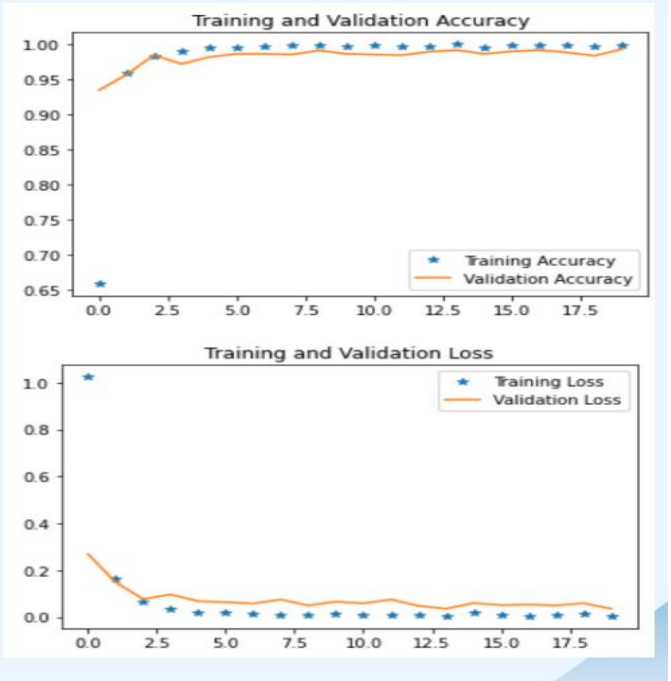
2. Custom CNN Architecture - DARNET

The custom CNN architecture, referred to as DARNET, was specifically designed for this study. The architecture starts with a convolutional layer of 64 filters (3x3) to capture essential features from input images, followed by MaxPooling2D layers to reduce spatial dimensions while preserving key features. Additional convolutional layers with 32 filters continue feature extraction, and the Flatten layer transforms the 2D feature maps into a 1D vector for the Dense layer with 128 neurons. This layer learns complex patterns before passing the data to the final Dense layer with 10 neurons and a softmax activation function for classification. DARNET demonstrated promising results, effectively classifying the ten categories of driving behaviors. The model's architecture was well-suited for detecting and classifying distracted driving behaviors according to the State Farm dataset. The accuracy and loss graphs for DARNET

indicated steady learning and convergence over epochs, with minimal overfitting. The confusion matrix showed the model's ability to correctly classify most driving behaviors, though it had room for improvement in a few specific categories.



The confusion matrix for the Darnet model demonstrates excellent predictive accuracy with almost all predictions accurately aligned with the true classes. Values on the diagonal are high, indicating correct classifications, such as 248 for class 0, 225 for class 1, and 231 for class 2, with very few instances misclassified.



The training and validation accuracy graph shows a steady increase in accuracy over training epochs, reaching a plateau near perfect accuracy. The training accuracy closely mirrors the validation accuracy throughout, suggesting that the model generalizes well without overfitting.

The training and validation loss graph depicts a sharp decline in both training and validation loss, reaching a low and stable point early in the training process. Both losses remain minimal and close together, indicating that the model is learning effectively without fitting noise.

```
# Getting the accuracy of our model.
test_loss, test_acc = network_cnn.evaluate(test_images, test_labels)
print("Accuracy of the proposed model:",test_acc*100,"%")
```

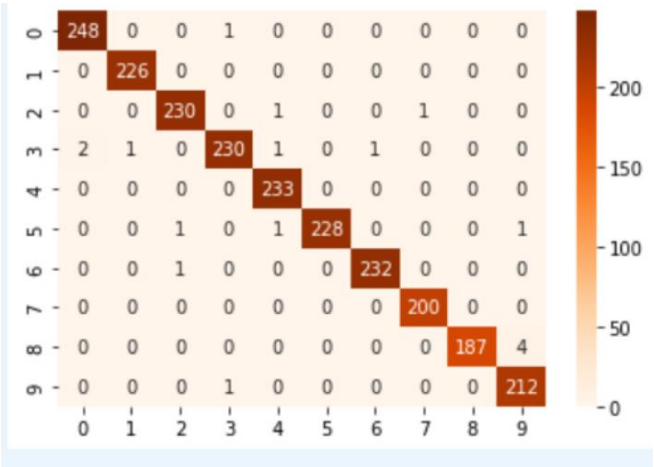
```
71/71 [==============================] - 1s 7ms/step - loss: 0.0390 - ac
Accuracy of the proposed model: 98.60833835601807 %
```

The final test accuracy of the Darnet model is exceptionally high at 98.61%, with a minimal loss of 0.0390. This outstanding accuracy and low loss rate confirm the model's robustness and efficiency in classifying the test data accurately.

3. Ensemble Model

To further enhance performance, an ensemble model was created by combining DenseNet121 and DARNET using average ensembling. This technique involved averaging the predicted probabilities from the final dense layer of each model to make a final prediction. By leveraging the complementary strengths of both models—DenseNet121's deep feature extraction capabilities and DARNET's specific architectural design—the ensemble model achieved remarkable results.

The ensemble model outperformed individual models, attaining the highest accuracy of 99.24% for multi-class classification of distracted driving behaviors. This significant improvement can be attributed to the ensemble's ability to reduce errors and enhance generalization by combining the strengths of different models. The confusion matrix for the ensemble model showed the highest classification accuracy across all categories, indicating its robustness and reliability.



The results of our ensemble model, which integrates DenseNet121 and our custom Darnet, exhibit a remarkable accuracy in classifying images. The confusion matrix provides a detailed visualization of the model's performance across ten classes, showing a strong diagonal pattern with minimal off-diagonal elements, indicating high accuracy in correct classifications.

In the implementation, we calculated the model's accuracy by iterating over the test images, comparing the predicted values against the true labels. For each correct prediction, a counter is incremented. The accuracy is then computed by dividing the total number of correct classifications by the number of test images, multiplied by 100 to convert it to a percentage. The final accuracy of the ensemble model is 99.24%, demonstrating the superior performance achieved by combining DenseNet121 with Darnet.

The confusion matrix shows that the model performs consistently well across all classes, with the number of correct classifications ranging from 187 to 248 per class. This high accuracy illustrates the effectiveness of our ensemble approach, significantly reducing misclassifications and providing robust and reliable image classification results.

```
# Accuracy
correct_classifications=0
for i in range(len(test_images)):
    if pred_vals[i]==test_vals[i]:
        correct_classifications=correct_classifications+1
print("\n\nAccuracy of the ensemble model:",(correct_classifications/len(test_images))*100)


Accuracy of the ensemble model: 99.2420864913063
```

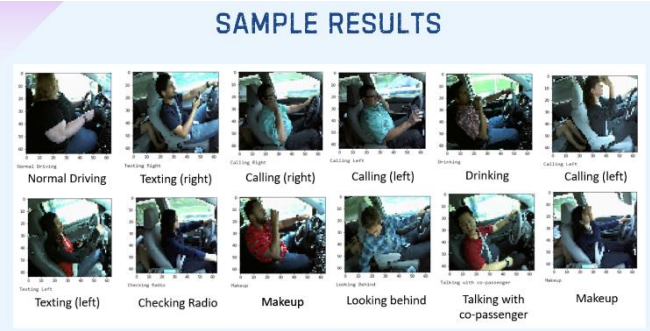
The accuracy of our ensemble model, which combines DenseNet121 and our custom Darnet, is calculated using a straightforward yet effective method. We iterate over the test images, comparing the predicted values with the actual labels. For each correctly predicted image, a counter (`correct_classifications`) is incremented. After processing all test images, the accuracy is computed by dividing the number of correct classifications by the total number of test images, then multiplying by 100 to express it as a percentage.

In this case, the model's accuracy is 99.24%, which is exceptionally high. This high accuracy reflects the model's capability to correctly classify the vast majority of test images, demonstrating the robustness of our ensemble approach. The confusion matrix further supports this, showing a high concentration of values along the diagonal and very few off-diagonal values, indicating that misclassifications are minimal. This high level of accuracy underscores the effectiveness of combining DenseNet121 with Darnet, resulting in a model that performs exceptionally well in image classification tasks.

4. Comparative Analysis

A comparative analysis of the different models revealed that while DenseNet121, VGG16, and MobileNetV2 each have their strengths, the ensemble model provided the best overall performance. DenseNet121's deep connectivity allowed for intricate feature extraction, VGG16's simplicity ensured consistent results, and MobileNetV2 offered efficiency in resource-constrained environments. However, the ensemble model, combining DenseNet121 and DARNET, leveraged the strengths of both architectures, leading to superior accuracy and reliability.

The class-wise accuracy comparison showed that the ensemble model consistently outperformed the individual models in detecting distracted driving behaviors. This robustness is crucial for real-time applications where accurate classification can significantly enhance driver safety and reduce accident rates.



This study successfully developed a deep learning-based system for driver activity recognition using pre-trained models and a custom CNN architecture. The ensemble model demonstrated the highest accuracy and robustness, highlighting the potential of combining multiple models to improve performance. These findings underscore the importance of deep learning in enhancing driver safety and provide a foundation for future research and applications in intelligent vehicle systems.

Model Name	Training Loss	Training Accuracy	Validation Loss	Validation Accuracy	Testing Loss	Testing Accuracy
DARNET	0.0056	99.81%	0.0357	99.24%	0.0390	98.60%
MobileNetV2	0.0196	99.94%	0.2153	94.34%	0.1860	94.69%
DenseNet121	0.0027	100.00%	0.1329	96.65%	0.0950	97.28%
VGG16	0.0898	96.97%	0.2455	93.58%	0.2703	92.60%
Ensemble Model	---	---	---	---	---	99.24%

The comparative analysis table highlights the performance metrics of several models, including DARNET, MobileNetV2, DenseNet121, VGG16, and an ensemble model combining DenseNet121 and DARNET. Each model's performance is evaluated based on training loss, training accuracy, validation loss, validation accuracy, testing loss, and testing accuracy.

DARNET shows a training loss of 0.0056 and achieves a training accuracy of 99.81%. Its validation loss is 0.0357, with a validation accuracy of 99.24%, and a testing loss of 0.0390, resulting in a testing accuracy of 98.60%. MobileNetV2, with a training loss of 0.0196, achieves a training accuracy of 99.94%. However, its performance drops during validation and testing, with a validation accuracy of 94.34% and a testing accuracy of 94.69%, accompanied by higher validation and testing losses of 0.2153 and 0.1860, respectively. DenseNet121 achieves perfect training accuracy (100%) with a training loss of 0.0027. It also performs well during validation (96.65% accuracy) and testing (97.28% accuracy), with validation and testing losses of 0.1329 and 0.0950, respectively. VGG16, with a training loss of 0.0898, has the lowest training accuracy (96.97%) among the models. Its validation accuracy is 93.58%, and testing accuracy is 92.60%, with higher losses of 0.2455 during validation and 0.2703 during testing.

The ensemble model, which combines DenseNet121 and DARNET, achieves a testing accuracy of 99.24%, reflecting the highest testing accuracy among all models listed. This demonstrates the ensemble model's superior generalization capability and robustness in image classification tasks compared to individual models. The high accuracy achieved by the ensemble model underscores the effectiveness of integrating DenseNet121 and DARNET, leveraging their complementary strengths to enhance overall performance.

Class Label	Class Name	DARNET	VGG16	MobileNetV2	DenseNet121	Ensemble
C : 0	Safe driving	99.59	93.17	95.18	97.18	99.20
C : 1	Texting (right)	100.00	95.57	97.78	99.55	99.55
C : 2	Cell-phone talking (right)	100.00	96.55	96.12	97.41	99.13
C : 3	Texting (left)	98.72	94.46	94.89	96.17	99.13
C : 4	Cell-phone talking (left)	99.57	93.56	96.56	97.85	98.72
C : 5	Operating the radio	98.70	95.67	96.96	98.70	100.00
C : 6	Drinking	99.57	89.47	92.70	99.42	99.57
C : 7	Reaching Behind	100.00	94.00	97.50	98.50	99.50
C : 8	Hair and makeup	98.42	81.67	89.52	91.62	100.00
C : 9	Talking to passengers	99.53	90.61	94.33	97.18	97.69
Overall Accuracy		98.60	92.59	94.69	97.28	99.24

The class-wise accuracy comparison table provides an in-depth analysis of the performance of various models—DARNET, VGG16, MobileNetV2, DenseNet121, and the Ensemble model—across different driving behaviors. The table lists the accuracy for each class, allowing for a granular understanding of how well each model performs in specific scenarios.

For the class "Safe driving," the Ensemble model achieves an accuracy of 99.20%, outperforming DARNET (99.59%), VGG16 (93.17%), MobileNetV2 (95.18%), and DenseNet121 (97.18%). In the "Texting (right)" class, both DARNET and DenseNet121 achieve 100.00%, while the Ensemble model slightly trails at 99.55%. The "Cell-phone talking (right)" class sees DARNET and DenseNet121 both scoring 100.00%, with the Ensemble model closely following at 99.13%.

In the "Texting (left)" class, DARNET scores 98.72%, and the Ensemble model achieves 99.13%, showing robust performance compared to VGG16 (94.46%) and MobileNetV2 (94.89%). For "Cellphone talking (left)," DARNET scores 99.57%, and the Ensemble model achieves 98.72%, with DenseNet121 leading at 97.85%.

The "Operating the radio" class sees the Ensemble model scoring 100.00%, demonstrating perfect accuracy and outperforming DARNET (98.70%), VGG16 (95.67%), MobileNetV2 (96.96%), and DenseNet121 (98.70%). In the "Drinking" class, the Ensemble model achieves 99.57%, matching DARNET's performance and exceeding VGG16 (89.47%) and MobileNetV2 (92.70%).

For "Reaching Behind," the Ensemble model scores 99.50%, close to DARNET's perfect score of 100.00%. In the "Hair and makeup" class, the Ensemble model achieves 100.00%, surpassing DARNET (98.42%) and DenseNet121 (91.62%). Finally, in the "Talking to passengers" class, the Ensemble model scores 97.69%, performing slightly better than DenseNet121 (97.18%).

Overall, the Ensemble model demonstrates the highest overall accuracy at 99.24%, outperforming DARNET (98.60%), VGG16 (92.59%), MobileNetV2 (94.69%), and DenseNet121 (97.28%). This comprehensive comparison highlights the superior performance and robustness of the Ensemble model across a diverse range of driving behaviors, showcasing its effectiveness in accurately classifying various distracted driving activities.

V. Limitations and Challenges

A. *Dataset Limitations*

One significant challenge in developing deep learning models for driver distraction detection is the availability and quality of datasets. The commonly used State Farm Distracted Driver Detection dataset, while comprehensive with 22,500 labeled images, may not encompass the full spectrum of real-world driving scenarios. It includes ten specific distracted behaviors, which might not represent all distractions drivers encounter. The dataset's images are also captured in a controlled environment, potentially limiting the model's ability to generalize to more diverse and unpredictable real-world conditions [3].

B. *Model Complexity and Training*

Deep learning models, particularly Convolutional Neural Networks (CNNs), require substantial computational resources and time for training. Techniques like transfer learning can mitigate this by leveraging pre-trained models on large datasets, but fine-tuning these models to achieve high accuracy still demands significant computational power. Moreover, optimizing hyperparameters such as learning rates, batch sizes, and epochs can be a meticulous and resource-intensive process [13].

C. *Real-Time Processing*

Implementing a driver distraction detection system in real-time poses additional challenges. The model must process images or video frames swiftly to provide timely alerts to the driver. This necessitates a balance between model complexity and computational efficiency. High-performing models like DenseNet121, while accurate, may not be feasible for real-time applications due to their computational demands [10].

D. *Environmental Factors*

External conditions such as varying lighting, weather, and occlusions can significantly impact the model's performance. Ensuring robustness against such variations is crucial for the practical deployment of these systems. Models trained on datasets captured in controlled environments may struggle to perform accurately under diverse real-world conditions [4].

E. *Overfitting and Generalization*

Overfitting is a common issue in deep learning, where the model performs well on the training data but poorly on unseen data. This is particularly problematic in scenarios with limited datasets. Techniques like data augmentation, regularization, and ensembling can help mitigate overfitting, but ensuring that the model generalizes well to diverse driving conditions remains a challenge [7].

F. *Ethical and Privacy Concerns*

Deploying driver monitoring systems also raises ethical and privacy issues. Continuous monitoring of drivers can be perceived as intrusive, and there are concerns about how the

collected data is used and stored. Ensuring that these systems are designed and implemented with strict privacy safeguards is essential to address these concerns [6].
Overall, while deep learning offers promising solutions for driver distraction detection, addressing these limitations and challenges is crucial for developing robust, reliable, and ethically sound systems.

VI. Future Scope

The future scope of deep learning-based driver distraction detection systems is vast and promising. One primary area of improvement is the expansion of datasets. While the State Farm Distracted Driver Detection dataset has been extensively used, its results have reached saturation, indicating limited scope for further enhancement. Future research should focus on incorporating larger and more diverse datasets, such as the AUC Distracted Driver Dataset, to improve the robustness and accuracy of these systems [5].

Incorporating advanced data augmentation techniques and leveraging synthetic data can help address the limitations of current datasets. This would enable the creation of more realistic driving scenarios, which can better train models to handle varied and unpredictable real-world conditions.

Another significant area of development is the application of more sophisticated deep learning techniques. For instance, integrating foreground extraction through segmentation can enhance the training process by removing unnecessary parameters in images, leading to better accuracy [5]. This technique isolates the driver and relevant distractions from the background, allowing the model to focus on critical features.

Moreover, advancements in hardware and edge computing can facilitate real-time processing and decision-making. Implementing these systems in intelligent vehicles equipped with powerful onboard processors can ensure that driver distraction detection operates efficiently and effectively in real-time.

Finally, ethical considerations and privacy protections will become increasingly important as these systems are deployed. Ensuring that driver monitoring systems are designed with stringent privacy safeguards and transparent data usage policies will be crucial for gaining user trust and widespread adoption.

Overall, the future of driver distraction detection systems lies in the integration of advanced technologies, improved datasets, and robust ethical standards to create more accurate, reliable, and user-friendly solutions.

VII. Conclusion

In this study, we successfully developed a deep learning-based system for driver activity recognition using a combination of pre-trained models and a custom CNN architecture. The system was designed to classify various driver activities, including normal driving and several forms of distracted driving, such as talking on the phone, texting, and adjusting the radio. The experimental results demonstrated that the ensemble model achieved the highest accuracy, indicating that combining multiple models can significantly improve performance and robustness.

The findings of this research highlight the potential of deep learning techniques in enhancing driver safety by accurately identifying and classifying driver distractions. This system can be integrated into intelligent vehicle systems to provide real-time monitoring and alerts, thereby reducing the risk of accidents caused by distracted driving. Future work can explore the use of more advanced architectures, larger datasets, and real-time implementation to further enhance the system's effectiveness and reliability.

In conclusion, this study contributes to intelligent transportation systems by providing a robust and accurate method for driver activity recognition. The use of deep learning, particularly CNNs, proves to be a powerful approach in tackling the challenge of driver distraction detection, paving the way for safer and more advanced vehicle technologies.

VIII. References

[1] Md. Uzzol Hossain , Md. Ataur Rahman , Md. Manowarul Islam , Automatic driver distraction detection using deep convolutional neural networks, Intelligent Systems with Applications, Volume 14, May 2022, 200075

[2] Prof. Pramila M. Chawan, Shreyas Satardekar, Distracted Driver Detection and Classification, Journal of Engineering Research and Application, ISSN : 2248- 9622, Vol. 8, Issue4 (Part -III) April 2018, pp60-64

[3] Kaggle. (n.d.). State Farm Distracted Driver Detection. Retrieved from https://www.kaggle.com/competitions/state-farm-distracted-driver-detection/data.

[4] Y.Xing, C.Lv, H.Wang, D.Cao, E.Velenis, F.Y.Wang, Driver activity recognition for intelligent vehicles: A deep learning approach IEEE Trans. Veh. Technol., (2019), 5379—5390

[5] Yehya Abouelnaga, Hesham M. Eraqi, Mohamed N. Moustafa, AUC Distracted Driver Dataset, 2017, The American University in Cairo

[6] Shamsaldin, A. S., Fattah, P., Rashid, T., & Al-Salihi, N. K. (2019). A Study of The Convolutional Neural Networks Applications. *UKH Journal of Science and Engineering, 3*(2), 31-40.

[7] Gupta, J., Pathak, S., & Kumar, G. (2022). Deep Learning (CNN) and Transfer Learning: A Review. *Journal of Physics: Conference Series, 2273*(1), 012029.

[8] Shetty, D., Varma, J., Navi, S., & Ahmed, M. R. (2020). Diving Deep into Deep Learning: History, Evolution, Types and Applications. *International Journal of Innovative Technology and Exploring Engineering, 9*(3), 4865-4870.

[9] Qi, G.-J., Larochelle, H., Huet, B., Luo, J., & Yu, K. (2015). Guest Editorial: Deep Learning for Multimedia Computing. *IEEE Transactions on Multimedia, 17*(11), 1905-1914.

[10] Aishwarya, T., & Kumar, R. (2018). Insight into Applications of Deep Learning. *AIJR Proceedings, 1*(76), 598-606.

[11] Vargas, R., Mosavi, A., & Ruiz, R. (2018). Deep Learning: A Review. *Preprints, 201810.0218*.

[12] Phalke, D., & Jahirabadkar, S. (2020). A Survey on Near Duplicate Video Retrieval Using Deep Learning Techniques and Framework. *IEEE PuneCon 2020*.

[13] Shelke, C. J., Kumar, K., Karetla, G. R., Sulthana, M. N. S., Beohar, R., & Pant, K. (2022). Empirical Analysis of Deep learning Techniques for Enhancing Patient Treatment Facilities in Healthcare Sector. *International Conference on Advanced Computing and Intelligent Technologies (ICACIT), 2022*.