Distracted Driver Detection Using MobileNetV2 and EfficientNet-B0

Muhammad Hammad Mobin

Department of Computer Science,
Texas State University, San Marcos, USA.
Email: har98@txstate.edu
Course: Machine Learning for Applications

***Abstract*—Driving a car is a complex task that requires complete attention. Distracted driving, any activity that diverts attention from the road, is a significant cause of road traffic accidents. According to the CDC Motor Vehicle Safety Division, one in five car accidents is caused by a distracted driver, resulting in approximately 425,000 injuries and 3,000 deaths annually. Addressing this issue, our project focuses on detecting distracted driving behaviors using machine learning models.**

Using a dataset of 2D dashboard camera images provided by State Farm, this work aims to classify driver behaviors into various categories of distraction, such as texting, talking on the phone, or other inattentive actions. The project implements and compares multiple machine learning architectures, including MobileNetV2, EfficientNet-B0, and a simple Convolutional Neural Network (SimpleCNN), to evaluate their effectiveness in detecting distracted behaviors. By leveraging these advanced deep learning models, we aim to enhance road safety and reduce the risks associated with distracted driving. This research lays the groundwork for future improvements in automated driver behavior monitoring systems.

I. Introduction

Object detection has been a cornerstone problem in computer vision, evolving significantly since the introduction of convolutional neural networks (CNNs) in the 1980s, inspired by the pioneering work of Hubel and Wiesel [1]. Modern deep learning techniques and advancements in computational power have enabled highly accurate and efficient object detection systems.

This project focuses on detecting driver behaviors using advanced CNN architectures. The dataset consists of images captured in a controlled environment, depicting drivers engaged in various activities such as texting, eating, or safely driving. The goal is to classify these images into one of 10 predefined classes: *c0: safe driving, c1: texting (right), c2: talking on the phone (right), c3: texting (left), c4: talking on the phone (left), c5: operating the radio, c6: drinking, c7: reaching behind, c8: hair and makeup, and c9: talking to a passenger.*

A. Dataset Description

The dataset used in this project was specifically designed to capture real-world driver behavior in a controlled setting. It contains images of drivers engaged in various activities that may distract their attention from the road, which is critical for developing safety systems that can monitor and alert drivers in real time. Each image is labeled according to the type of behavior exhibited by the driver, enabling the training of a model that can accurately classify these behaviors.

The dataset is well-balanced, with approximately equal representation for each of the 10 predefined classes, ensuring that the model learns to distinguish between different behaviors effectively. These images were captured under varying lighting conditions to simulate real-world scenarios, contributing to the robustness of the model. The dataset's relatively small size makes it suitable for transfer learning techniques, allowing the use of pre-trained models and fine-tuning them to this specific task.

The methodology involves training the models on labeled driver behavior data to classify each image accurately. By leveraging transfer learning and fine-tuning techniques, the models are adapted to this specific problem domain. This approach ensures robust performance even with a relatively constrained dataset.

Fig. 1. Dataset

The layout of this report follows a structured format akin to a research paper. Section II provides a *Literature Review*, detailing related works in distracted driving detection and object classification. Section III outlines our methodology, including data preprocessing, model architecture, and training strategies. The results and evaluation metrics are discussed in Section IV, followed by conclusions and future work in Section V.

II. Literature Review

The primary goal of this project is accurate classification and detection of driver behavior in dashboard images using advanced convolutional neural networks (CNNs). This section reviews the key architectures and methodologies relevant to this domain, including MobileNet, EfficientNet, and their applications in image classification tasks.

A. MobileNet

MobileNet [2] is a family of lightweight and efficient deep neural network architectures designed for mobile and embedded vision applications. It introduces depthwise separable convolutions, which significantly reduce the number of parameters and computational costs without compromising accuracy. MobileNet models have been widely adopted in tasks requiring real-time performance on resource-constrained devices, including object detection, segmentation, and image classification. Given the resource-constrained nature of dashboard-mounted cameras, MobileNetV2 [3], which builds upon the original MobileNet, is an optimal choice. It incorporates inverted residuals and linear bottlenecks to improve the model's representational power while maintaining its efficiency. These features make MobileNetV2 particularly suitable for real-time distracted driving detection.

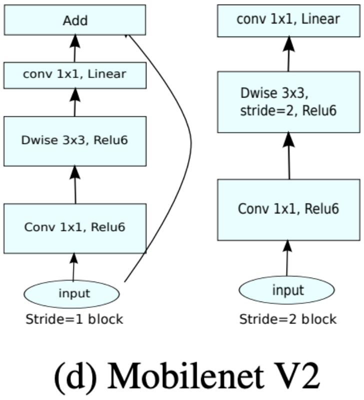

Fig. 2. Mobilenet Architecture

B. EfficientNet

EfficientNet [4] proposes a novel scaling method called compound scaling, which uniformly scales network width, depth, and resolution. This balanced approach enables EfficientNet models to achieve state-of-the-art accuracy while being computationally efficient. EfficientNet-B0, the baseline model in the EfficientNet family, is particularly suitable for tasks with limited computational resources while still delivering competitive performance. Its architecture leverages mobile inverted bottleneck convolution (MBConv) blocks and squeeze-and-excitation optimization to improve feature extraction. These attributes make it an excellent choice for driver behavior classification, enabling accurate detection of distractions such as texting, eating, or talking on the phone.

C. Data Augmentation Techniques

To improve model generalization and robustness, data augmentation is employed extensively during training. The augmentation pipeline includes resizing images to a consistent resolution, random horizontal flipping, and random rotations to simulate real-world variations. Additionally, normalization using ImageNet mean and standard deviation values ensures the input data aligns well with pre-trained models such as MobileNet and EfficientNet. These augmentations enable the model to effectively handle challenges such as varying lighting conditions and image orientations, which are critical for robust performance.

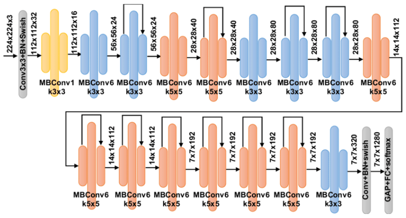

Fig. 3. EffecientNet Architecture

D. Evaluation Metrics and Benchmarking

Model evaluation is performed using a comprehensive set of metrics, including accuracy, precision, recall, and F1-score, calculated on the validation dataset. The F1-score, a harmonic mean of precision and recall, is defined as:

$$F1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \quad (1)$$

A confusion matrix is also generated to analyze the per-class performance, providing insights into misclassifications. For visualization, a heatmap of the confusion matrix highlights the model's strengths and areas for improvement across different classes. These evaluation metrics ensure a thorough assessment of the model's capability in classifying images accurately, and they serve as a benchmark for comparing performance across different architectures.

III. Methodology

We implement multiple architectures to evaluate their performance for the image classification task. This section details the models used, including UNet, SimpleCNN, MobileNetV2, and EfficientNet. Each model is trained with specific hyperparameters and methodologies to optimize their performance.

A. SimpleCNN

The SimpleCNN model is a custom convolutional neural network designed for lightweight image classification tasks. The architecture consists of:

- A single convolutional layer with 32 filters, kernel size of 3, and stride of 1, followed by ReLU activation.
- Max pooling with a pool size of 2.
- A fully connected layer with 256 units, followed by another ReLU activation.
- An output layer with a number of units equal to the number of classes.

The model is trained with a batch size of 32 using the Adam optimizer, with an initial learning rate of 0.001. Cross-entropy loss is used as the training objective. This simple architecture provides a baseline for performance comparison against more complex models.

B. MobileNetV2

MobileNetV2, a pre-trained lightweight neural network, is fine-tuned for this task. The architecture leverages depthwise separable convolutions to reduce computational complexity. The final classification layer is modified to match the number of target classes. Training details include:

- Optimizer: AdamW with an initial learning rate of 0.0001.
- Batch size: 32.
- Loss function: Cross-entropy loss.

Augmentation techniques are applied, ensuring robustness to variations in the dataset. MobileNetV2 achieves a balance between performance and computational efficiency, making it ideal for scenarios requiring faster inference.

C. EfficientNet

EfficientNet, specifically the `efficientnet-b0` variant, is employed for its state-of-the-art performance and efficiency. The model is pre-trained on ImageNet and fine-tuned by replacing the final classification layer to match the target classes. Training details include:

- Optimizer: AdamW with an initial learning rate of 0.0001.
- Batch size: 32.
- Loss function: Cross-entropy loss.

EfficientNet incorporates compound scaling to optimize network depth, width, and resolution. Augmentations such as resizing, rotations, and flips are applied to the training data to improve generalization.

D. Training and Evaluation Protocol

All models are trained using PyTorch Lightning to streamline the training pipeline. During training, the loss and other metrics (e.g., accuracy, precision, recall, and F1-score) are logged for monitoring. Validation is performed after each epoch to ensure the model's performance generalizes well to unseen data. Augmentations are consistent across models to ensure a fair comparison.

The methodology ensures a thorough evaluation of architectures ranging from simple custom designs to complex pre-trained models, catering to a variety of computational and application-specific requirements.

IV. Results

A. SimpleCNN

For the SimpleCNN model, the overall accuracy achieved was 90.99%. The model exhibited excellent performance in terms of precision, recall, and F1-Score, with values of 0.9113, 0.9105, and 0.9094, respectively. The model also demonstrated high accuracy across various driving scenarios:

- Safe driving: 86.80%
- Texting (right): 95.63%
- Talking on the phone (right): 83.01%
- Texting (left): 96.73%
- Talking on the phone (left): 95.36%
- Operating the radio: 95.95%
- Drinking: 88.61%
- Reaching behind: 92.72%
- Hair and makeup: 86.75%
- Talking to passenger: 88.89%

Figure 4 shows the ROC curve for the SimpleCNN model, highlighting its strong classification capabilities.

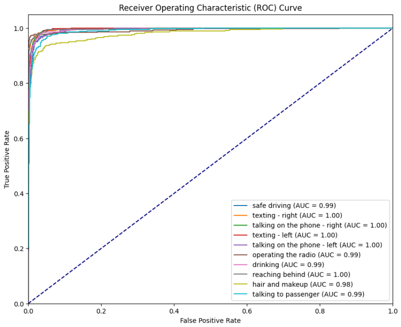

Fig. 4. ROC Curve for SimpleCNN Model

B. MobileNet

The MobileNet model achieved a remarkable overall accuracy of 99.73%. The precision, recall, and F1-Score were equally impressive, with all values surpassing 99%. Here are the accuracies for different driving activities:

- Safe driving: 99.61%
- Texting (right): 99.78%
- Talking on the phone (right): 100%
- Texting (left): 100%
- Talking on the phone (left): 99.77%
- Operating the radio: 98.93%
- Drinking: 100%
- Reaching behind: 100%
- Hair and makeup: 99.48%
- Talking to passenger: 99.76%

The ROC curve for the MobileNet model is shown in Figure 5, demonstrating its exceptional performance across various categories.

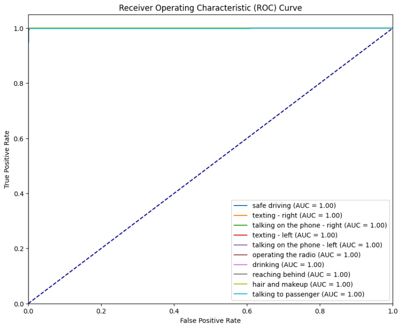

Fig. 5. ROC Curve for MobileNet Model

C. *EfficientNet*

The EfficientNet model also performed exceptionally well, achieving an accuracy of 99.35%. Its precision, recall, and F1-Score were all very close, with values of 0.9935, 0.9931, and 0.9932, respectively. The model demonstrated near-perfect performance in the following categories:

- Safe driving: 100%
- Texting (right): 100%
- Talking on the phone (right): 100%
- Texting (left): 99.78%
- Talking on the phone (left): 99.88%
- Operating the radio: 99.57%
- Drinking: 99.67%
- Reaching behind: 100%
- Hair and makeup: 99.73%
- Talking to passenger: 99.68%

Figure 6 displays the ROC curve for EfficientNet, showcasing its robustness across all categories.

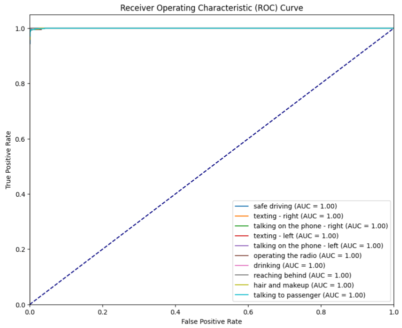

Fig. 6. ROC Curve for EfficientNet Model

V. Conclusion and Future Work

This study has provided a foundation for exploring various deep learning models and training methodologies for image classification and object detection tasks. There are several avenues for future research and development, including:

1) **Exploring New Networks:**
   - Experiment with models like Tiny YOLO to achieve real-time performance in object detection tasks.
   - Investigate other efficient models for constrained environments.
2) **Building a Real-World Application:**
   - Develop a mobile application using `Streamlit` to enable:
     - Periodic image capture.
     - Real-time evaluation of distraction behavior.
   - Extend the app to include user-friendly interfaces for monitoring and reporting.
3) **Advanced Analysis:**
   - Implement video-based detection for continuous monitoring, enhancing accuracy and robustness.

These extensions aim to bridge the gap between theoretical experimentation and real-world applications, pushing the boundaries of efficiency and accuracy in image processing and object detection systems.

References

[1] K. Fukushima. “Neocognitron”. In: *Scholarpedia* 2.1 (2007). revision #91558, p. 1717. DOI: 10 . 4249 / scholarpedia.1717.

[2] Andrew G Howard et al. “MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications”. In: *arXiv preprint arXiv:1704.04861* (2017).

[3] Mark Sandler et al. “MobileNetV2: Inverted Residuals and Linear Bottlenecks”. In: *arXiv preprint arXiv:1801.04381* (2018).

[4] Mingxing Tan and Quoc V Le. “EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks”. In: *arXiv preprint arXiv:1905.11946* (2019).