# Distracted Driver Detection Using Deep Learning

Pranay Reddy Gundala[1], Srikar Devulapalli[2], Trevor Peyton[3], and Ruth Balaji[4]

[1] e-mail: vgundala@iu.edu
[2] e-mail: vdevula@iu.edu
[3] e-mail: trpeyton@iu.edu
[4] e-mail: rutbala@iu.edu

November 30, 2023

## ABSTRACT

This study addresses the issue of distracted driving, a major cause of vehicular accidents, by developing a Deep Learning Classifier. This model, trained on images from dashboard cameras, aims to identify distracted driving behaviors. It compares a baseline model with convolutional layers against advanced models using pre-trained weights from ResNet, VGG, and MobilNet. Our objective is to determine the most effective model for enhancing road safety by detecting and reducing distracted driving incidents.

## 1. Introduction

The detection of distracted drivers is a critical aspect of modern technological advancements aimed at enhancing road safety. Given the significant risks posed by distracted driving, it has emerged as a pressing global issue. Common distractions include texting, smartphone usage, eating, adjusting the radio, and other activities that divert attention from driving. In response to this challenge, systems for detecting distracted drivers have been increasingly implemented. These systems utilize deep learning models to continuously monitor and analyze driver behavior. To effectively identify signs of distraction, they often integrate a combination of sensors, cameras, and sophisticated artificial intelligence (AI) algorithms.

## 2. Methodology

In this section we will discuss the dataset and variousdeep learning architectures we experimented with.

### *2.1.* Dataset

Kaggle, an online platform for data science competitions, offers datasets for cutting-edge algorithmic challenges in sectors less exposed to AI advancements. It provides a platform to understand datasets deeply and offers opportunities like internships and awards for students. We utilize a dataset from a completed State Farm challenge, which includes hundreds of images depicting various actions performed by multiple individuals.

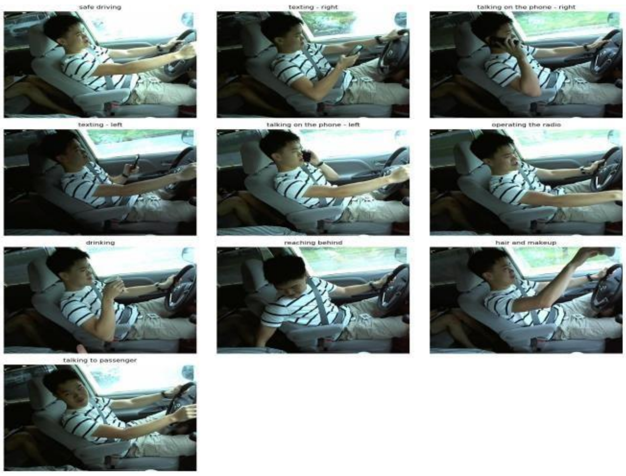

*Figure 1. Images of our dataset classes*

Figure 1, presents a classification of distracted driving behaviors, including texting (right and left hand), talking on the phone (right and left hand), operating the radio, drinking, reaching behind, adjusting hair or makeup in the mirror, and attentive driving.

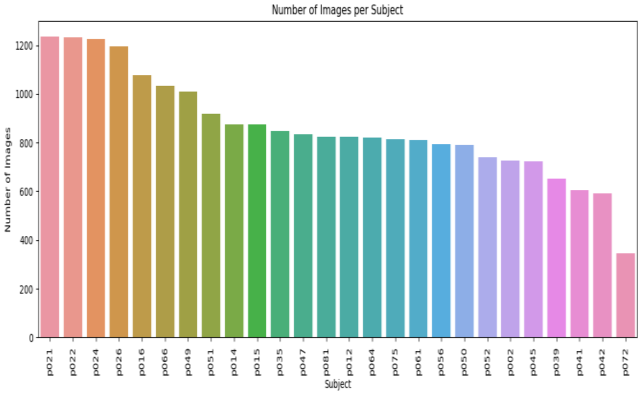

*Figure 2. Number of Images per subject*

The dataset encompasses images from 26 individuals, each contributing between 400 to 1200 images across different classes. Figure 2, illustrates the distribution of images per subject, highlighting a notable variance, such as the 800-image difference between subjects p072 and p021. Despite this disparity, we retained the distribution to assess its impact on the results, observable through a confusion matrix.

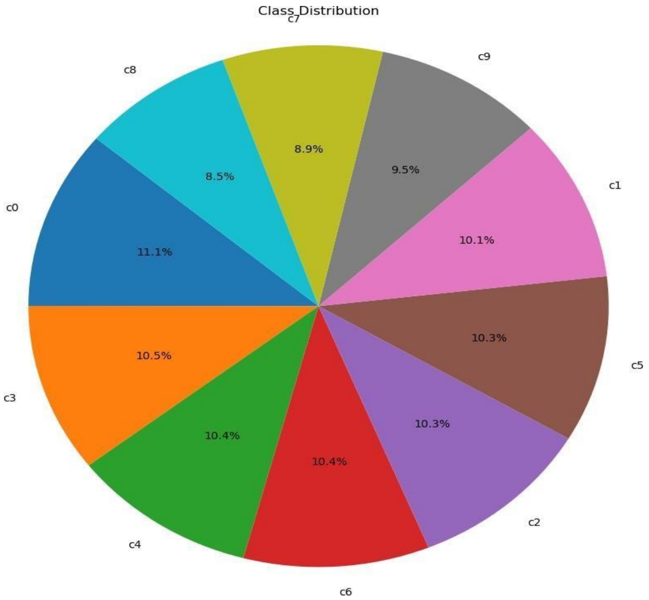

*Figure 3. Distribution of classes*

Figure 3, depicts the overall class distribution among users, crucial for identifying potential biases in the dataset. Although some classes have slightly lower representation, the distribution is balanced.

## *2.2.* Baseline Model

Our initial approach involved creating a baseline model to assess the learning capabilities from scratch. This model comprised four convolutional layers with 32, 64, 128, and 256 filters, respectively. Following each convolutional layer, a ReLU activation function introduces a nonlinear component, opting for the maximum value between the neuron's output and zero. Subsequent to ReLU, we apply a 2x2 MaxPoling layer, significantly reducing computational complexity by retaining only maximum values. Batch normalization follows, ensuring normalized outputs relative to the entire batch. Each convolution layer concludes with a 10% dropout to promote model generalization and mitigate overfitting (Alpert, n.d.).

The architecture then flattens the convolution outputs into a single vector per batch, passing through a 512-neuron linear layer with ReLU activation and dropout. The final layer matches the number of distinct driver actions, summing up to 2,493,130 trainable parameters.

## *2.3.* MobileNet

MobileNet employs depth-wise convolution layers, which feature distinct filters for each input channel, reducing computational demand. A depth-wise layer is paired with a pointwise layer (a standard 1-kernel size convolution layer) to establish channel relationships while maintaining low computational costs.

## *2.4.* VGG (Visual Geometry Group)

VGG, particularly the VGG-50 variant used in our study, comprises 50 convolution layers, incorporating max pooling operations and ReLU activation functions. It represents a deep neural network approach with standard convolution layers.

## *2.5.* ResNet

ResNet addresses the overfitting issue seen in deep networks like VGG through residual blocks. These blocks include standard convolution layers and incorporate skip connections every few layers, allowing the networks to leverage outputs from previous layers alongside current ones*(Driver Distraction Detection Using Advanced Deep Learning Technologies Based on Images, 2022)*. This approach facilitates the learning of new features from additional contexts (Deep Residual Learning for Image Recognition, 2016).

## *2.6.* ResNet Hyperparameter Tuning

ResNet's underperformance was unexpected, given its state-of-the-art status and typical high accuracy. Attempts at manual hyperparameter tuning, including adjustments to the learning rate and batch sizes, did not yield significant improvements. Research indicated that ResNet's complexity, especially its extensive parameter count, might be excessive for our simpler task. Some studies suggested better performance with partial unfreezing of convolutional layer weights for task-specific retraining. However, we opted against this to maintain a consistent comparison across models and due to our limited resources. Consequently, we concluded that ResNet was not an optimal choice for our specific application.

## *2.7.* Transfer Learning

For MobileNet, VGG, and ResNet, we utilized transfer learning due to their extensive parameter sets. We imported pre-trained weights from models trained on the vast ImageNet database, freezing them to prevent updates during backpropagation. Our focus was on training the final fully connected layers, aligned with our baseline model's configuration. Leveraging ImageNet's learned features for precise predictions on our dataset.

## 3. Experimental Setup

For the implementation and training of our models, we utilize the TensorFlow library, chosen for its ease of use and efficiency. Given our computational resources and time constraints, we incorporated an early stopping strategy (Chen, 2021). Training would cease if test accuracy did not improve over four consecutive epochs. In instances of suboptimal model performance, we employed manual hyperparameter tuning to seek improvements.

Our training process was divided between multiple group members on various hardware. The final models shown in this report were trained on a Mac with an M1 accelerated GPU. It ran the TensorFlow-macOS version 2.15.0, and TensorFlow-metal version 1.1.0.

## 4. Result

All models were trained using the Adam optimizer, set at a learning rate of 0.001. We employed Sparse Categorial Cross entropy as our loss function, coupled with a SoftMax activation function in the final output layer. The use of 'Sparse" was dictated by our class representation, which was numeric rather than one-hot encoded.

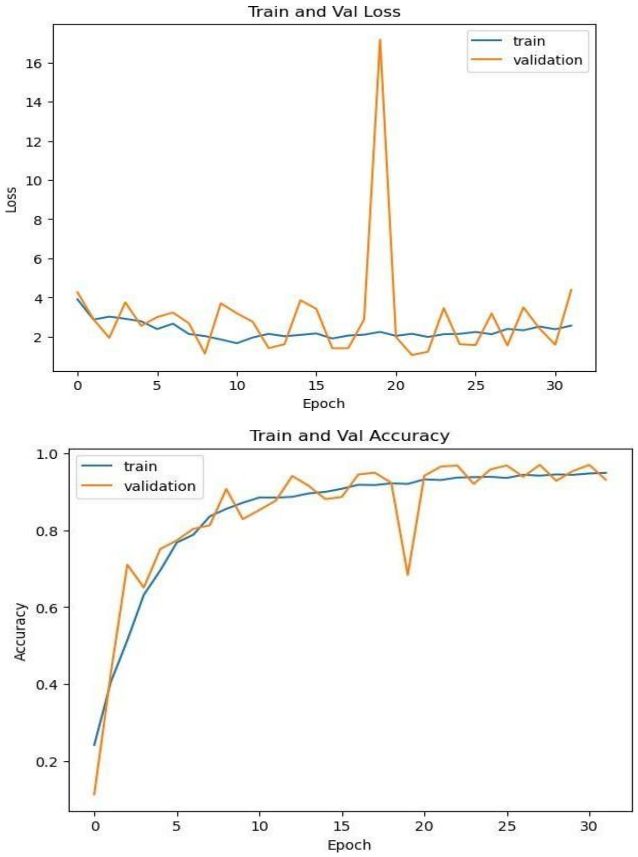

*Figure 4. Baseline results. First shows training and validation loss per epoch and second shows training and validation accuracy.*

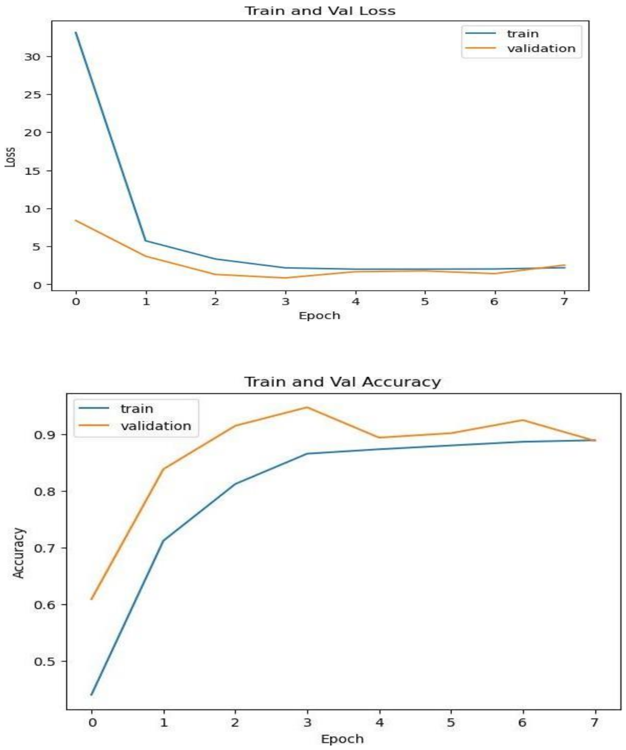

*Figure 5. MobileNet's results. First one shows training and validation loss per epoch, and second shows training and validation accuracy.*

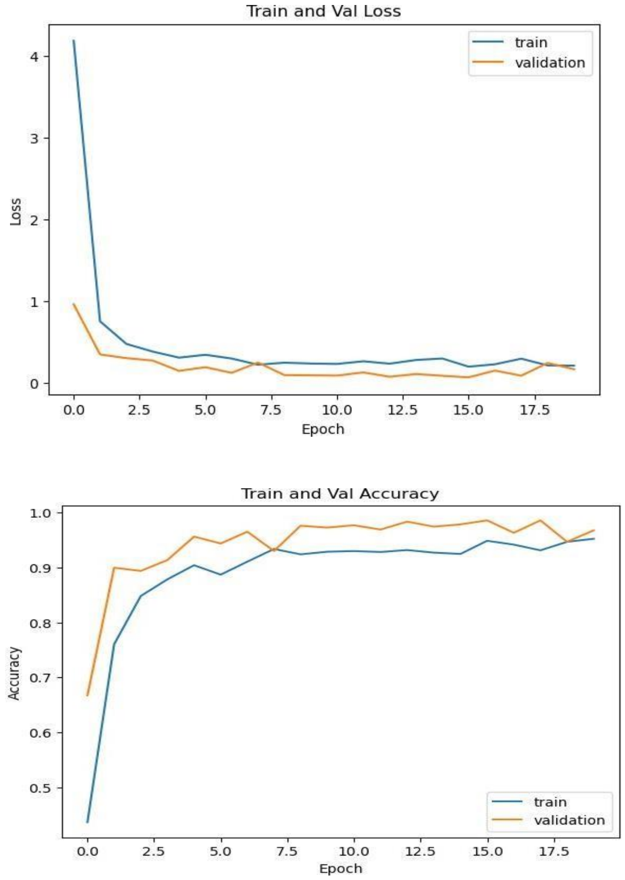

*Figure 6. VGG results. First shows training and validation loss per epoch and second shows training and validation accuracy.*

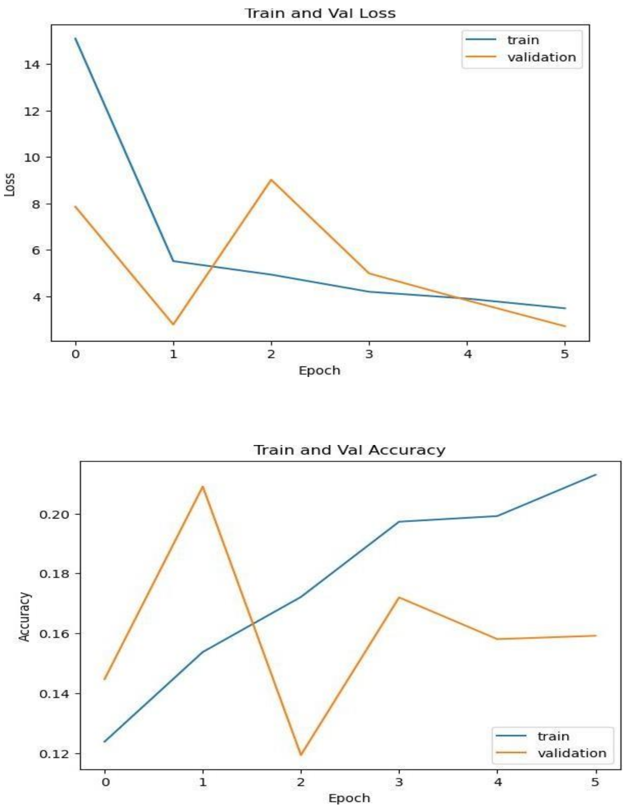

*Figure 7. ResNet results. First shows training and validation loss per epoch and second shows training and validation accuracy.*

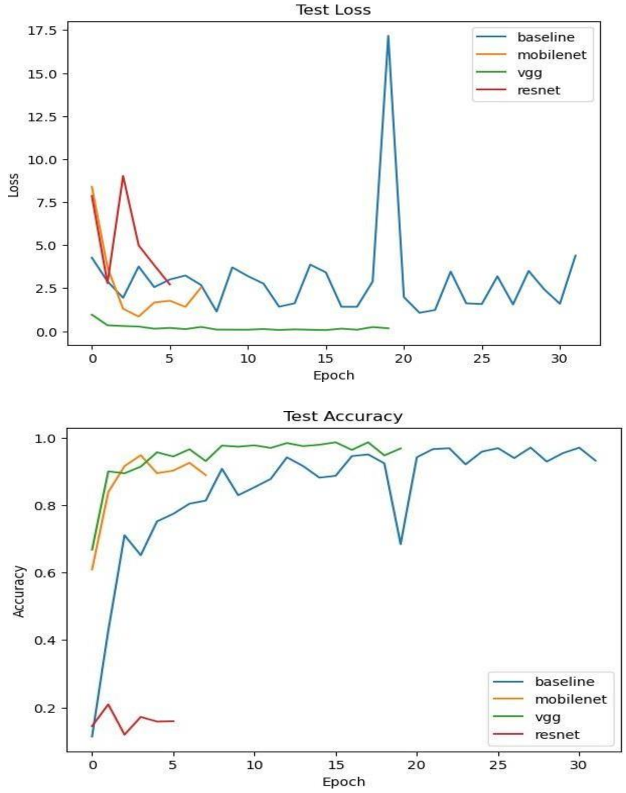

*Figure 8. Every model's result plotted together. First shows test loss per epoch, and second shows test accuracy.*

Figures 4 to 8 present the training and validation losses and accuracies for our Baseline, MobileNet, VGG, and ResNet models, respectively, with Figure 8 providing a comparative overview. Notably, ResNet, MobileNet, and VGG training were halted early due to stagnation over four epochs, while our baseline model continued for an additional 10 epochs. This extended training period is attributed to the baseline's larger parameter space allowing for incremental improvements.

The results demonstrated the feasibility of achieving over 95% accuracy in both training and testing. VGG emerged as the most effective model, marginally outperforming the baseline model, followed by MobileNet and ResNet. VGG's depth, even with fixed CNN features, provided sufficient high-level information for complex decision-making. Our baseline model, though less parameterized that VGG was specially trained for this task, resulting in high accuracy. MobileNet, while less accurate, offered a significant computational speed advantage, which could be crucial in real-time applications prioritizing frame rate over accuracy.

## 5. Conclusion

Distracted driving poses a significant threat on the roads, contributing to numerous fatalities annually. Through this study, we have demonstrated the efficacy of deep neural networks in accurately detecting driver distraction. The implications of this technology are profound (Kim et al., 2023). It paves the way for advanced, real-time systems in vehicles that can alert drivers when they are distracted for extended periods. Furthermore, this technology can be adapted for identifying drivers at risk of falling asleep at the wheel and issuing timely warnings.

The application of machine learning in addressing critical safety issues like distracted driving represents a major stride in fostering a safer environment for all road users. Our findings underscore the vital role of technological advancements in enhancing road safety and reducing the incidence of accidents and injuries.

## 6. Team Member's Contribution

For our group, everyone did a little bit of each part, and no one thing can be attributed to anyone, however, we broke up the major roles in the beginning as follows. Ruth would be in charge of the PowerPoint and paper formatting and creation. Srikar would be in charge of the data preprocessing and any steps to prepare it for the model. Pranay set up the model architectures and scripts in preparation for training. Finally, Trevor trained the models and provided post-analysis code and discussion.

## 7. References

[1] Driver Distraction Detection Using Advanced Deep Learning Technologies Based on Images. (2022). IEEE Journals & Magazine | IEEE Xplore. https://ieeexplore.ieee.org/document/9913644

[2] Kim, D., Park, H., Kim, T., Kim, W., & Paik, J. (2023, October 25). Real-time driver monitoring system with facial landmark-based eye closure detection and head pose recognition. Scientific Reports. https://doi.org/10.1038/s41598-023-44955-1

[3] Alpert, B. (n.d.). How Nauto Detects Distracted Drivers with Machine Learning | Nauto. https://www.nauto.com/blog/nauto-engineering-deep-learning-for-distracted-driver-monitoring

[4] Chen, B. (2021, December 15). Early Stopping in Practice: an example with Keras and TensorFlow 2.0. Medium. https://towardsdatascience.com/a-practical-introduction-to-early-stopping-in-machine-learning-550ac88bc8fd

[5] Deep Residual Learning for Image Recognition. (2016, June 1). IEEE Conference Publication | IEEE Xplore. https://ieeexplore.ieee.org/document/7780459