Drowsy Driver Detection System

Jayasudha Kalamegam, Keerthi Gajjela, Kritthika Shanmugam

Sathiyashivani Sathish Kumar,

TP02, DS620 Machine Learning/Deep Learning, MSCS,

City University of Seattle

[kalamegamjayasudha@cityuniversity.edu](mailto:kalamegamjayasudha@cityuniversity.edu)

[gajjelakeerthi@cityuniversity.edu](mailto:gajjelakeerthi@cityuniversity.edu)

[sathishkumarsathiya@cityuniversity.edu](mailto:sathishkumarsathiya@cityuniversity.edu)

[shanmugamkritthika@cityuniversity.edu](mailto:shanmugamkritthika@cityuniversity.edu)

**Abstract**

The goal of our project is to implement the Driver Drowsiness Detection System using machine learning and deep learning techniques which offers a user-friendly approach for identifying drowsy drivers through various image analyses. This project is well designed to be accessible for everyone even for beginners, providing a simple and straight analysis yet an effective solution. By utilizing a dataset taken from a Kaggle source which is a trustworthy site that has a collection of sleepy and alert individual's images, the system employs a convolutional neural network architecture trained to develop and identify slight facial expressions or features that capture the image of drowsiness such as eye closure, yawning, and head nodding. The project provides the users with the classification outcome by clearly indicating whether the individual in the uploaded image is in a state of drowsiness or alertness. To implement this project application, the OpenCV library, Keras, and ML algorithms have been used. This project works to save lives by preventing road accidents caused by driver drowsiness, showcasing a practical application of AI in addressing real-world safety issues.

**Keywords:** Driver Drowsiness Detection, Machine Learning, Computer Vision, Road Safety, User Interface Design, Accessibility.

1. **INTRODUCTION**

Drowsy driving poses a significant threat to road safety, counting to a number of accidents worldwide each year. This consequences of driver fatigue extend beyond property damage, often resulting in severe injuries or fatalities. Recognizing the urgent need for effective intervention, our project helps to implement a Driver Drowsiness Detection System leveraging machine learning and deep learning techniques. By harnessing the power of image analysis, our system aims to offer a user-friendly solution accessible to all, irrespective of expertise levels.

Through the utilization of a diverse dataset sourced from Kaggle, encompassing images of both sleepy and alert individuals, our system employs a convolutional neural network architecture trained to discern subtle facial expressions indicative of drowsiness, such as eye closure, yawning, and head nodding. Moreover, to enhance the robustness of our model, we implemented data augmentation techniques, including rotation, width and height shifts, shear, zoom, and horizontal flipping, ensuring better generalization to real-world scenarios.

By harnessing the power of artificial intelligence, particularly deep learning algorithms, the DDDS aims to accurately assess the alertness level of drivers in real-time based on facial cues. As, this approach we seek to provide users with a reliable tool for identifying drowsy drivers, thereby mitigating the risk of road accidents, and potentially saving numerous lives**.**

1. **LITERATURE REVIEW**
2. **METHODOLOGY**

Driver drowsiness poses a significant safety risk on roads globally, with fatigue often causing accidents. Accident detection methods have limitations, prompting exploration of computer vision and machine learning, particularly CNNs. These models can analyze facial expressions and physiological signals to identify drowsiness indicators accurately and non-intrusively. Despite progress, challenges persist in data quality, model robustness, and real-world implementation. Addressing these challenges and leveraging machine learning aims to develop accessible and reliable drowsiness detection systems, crucial for enhancing road safety.

**Real-Time Driver Drowsiness Detection System Using CNN:** The paper "Real-Time Driver Drowsiness Detection System Using Convolutional Neural Networks" by Xie, Y., Yu, H., Miao, X., et al. (2020) proposes a real-time driver drowsiness detection system using convolutional neural networks (CNNs). Trained in facial images from real driving scenarios, the CNN model recognizes drowsiness-related facial features like eye closure and head movements. Integrated with in-vehicle cameras, the system accurately detects drowsy driving behavior, offering a promising solution for enhancing road safety by leveraging CNN-based methods.

**Driver Drowsiness Detection System Based on Deep Learning Techniques:** The paper "Driver Drowsiness Detection System Based on Deep Learning Techniques: A Review and Future Directions" by Li, J., Guo, Y., Zhang, L., et al. (2021) comprehensively reviews recent advancements in deep learning for driver drowsiness detection, focusing on CNNs and RNNs. It discusses common datasets, evaluation metrics, challenges, and future research directions, offering valuable insights for researchers and practitioners in this field.

**Drowsiness detection for safety driving based on multimodal deep learning:** The paper by Zhenyu Wang et al. (2020) proposes a method to detect driver drowsiness using a combination of visual and physiological data analyzed through deep learning. By integrating information from sources like facial expressions and physiological signals, such as heart rate, the system aims to provide a more robust and accurate assessment of driver alertness. This approach offers the potential to enhance driving safety by alerting drivers before they reach a dangerous level of drowsiness.

**Model selection**: Convolutional Neural Networks (CNNs) are suitable for visual data like facial expressions and eye movements due to their ability to extract hierarchical features from images. By selecting the right models for each data modality, the system can effectively detect patterns associated with drowsiness, contributing to improved accuracy in classification.

**Dataset:** The driver drowsiness dataset consists of images categorized into four classes: Closed Eyes, Open Eyes, yawn and no yawn. The dataset is structured as follows:

**Closed Eyes:** 726 images

**Open Eyes:** 726 images

**Yawn:** 723 images

**No yawn:** 725 images

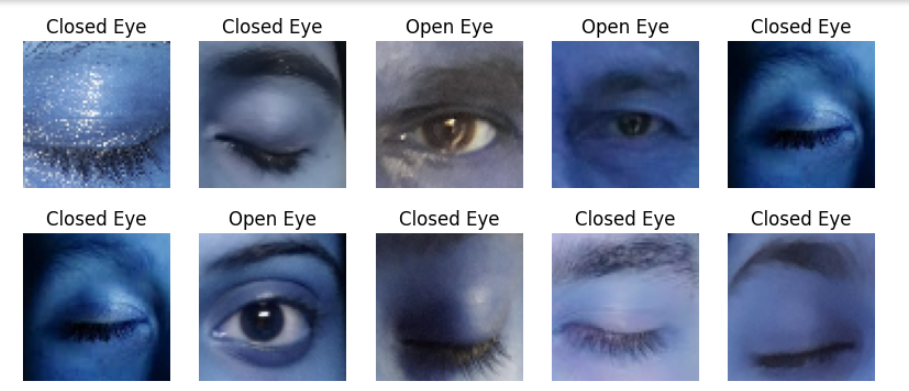
We chose closed and open eye images for driver drowsiness detection because eye state is a direct and reliable indicator of alertness. As, this binary classification simplifies the model design and enhances accuracy, leveraging clear visual cues and abundant datasets. This approach enables efficient real-time detection, crucial for preventing road accidents caused by drowsy driving. The dataset is used from kaggle and it is balanced, with an approximately equal number of images in both the category, which is used for effective training and evaluation of models for drowsiness detection. Since, dataset is uploaded as a zip file, we used zip file extract to upload in the Google colab. These images are stored in separate folders corresponding to their respective classes, making it easy to preprocess and label them. The dataset is chosen and used for binary classification tasks, such as detecting whether a driver's eyes are open or closed. This comprehensive dataset provides a solid foundation for developing machine learning models to detect driver drowsiness.

**Preprocessing:** The crucial part of the project is the preprocessing steps which is the preparing process of the input data for effective analysis by the convolutional neural network (CNN). Initially, we downloaded a diverse dataset containing images of both sleepy and alert individuals, sourced from Kaggle. Upon loading the dataset, the images undergo a series of preprocessing steps to ensure uniformity and suitability for model training and for getting better performance.

**Resizing:** we resize all images to a consistent size, typically 128x128 pixels, to facilitate uniform input dimensions for the CNN. This resizing step ensures that the model receives standardized input images, essential for maintaining consistency and optimizing computational efficiency during training.

**Normalization:** Following resizing, we normalize the pixel values of the images to a range between 0 and 1. Normalization aids in stabilizing the training process by bringing all input features to a similar scale, this preventing certain features from dominating others during model performance.

The training dataset images are displayed with their labels closed eyes and open eyes.



**Visualization:** The visualization component provides crucial insights into the performance and behavior of the drowsy driver detection model. Accuracy and loss curves offer a comprehensive view of the model's learning progress, while sample images with predictions allow for quick qualitative assessment of real-world performance. These visualizations aid in refining the model's accuracy and reliability for effective drowsy driver detection.

These preprocessing steps ensure that the input data are appropriately formatted and standardized setting the stage for effective training and accurate drowsiness detection by our CNN-based system.

1. **CNN ARCHITECTURE**

Convolutional Neural Networksare specialized types of artificial neural networks used primarily for processing grid-like data such as images. CNNs are designed to automatically and adaptively learn spatial hierarchies of features through backpropagation by using multiple building blocks, including convolutional layers, pooling layers, and fully connected layers.

A diagram of a layer

Description automatically generated

**Key Components of CNN Architecture:**

**Input Layer**: The layer where the raw input data (e.g., an image) is fed into the network. For example, an image with dimensions 32x32x3 (width x height x depth, where depth represents color channels like RGB).

**Convolutional Layers**: Apply a set of learnable filters (kernels) to the input image to extract features. Each filter slides (convolves) across the image, computing dot products between the filter weights and the image patch it covers. The output is a set of feature maps, each representing different features of the input.

**Activation Layer**: Introduces non-linearity to the model by applying an activation function elementwise to the output of the convolution layer. Common activation functions include ReLU (Rectified Linear Unit), Tanh, and Leaky ReLU. Ensures the network can model complex patterns.

**Pooling Layer**: Reduces the spatial dimensions (width and height) of the feature maps, retaining the most significant information. Types of pooling include Max Pooling (selects the maximum value from each patch of the feature map) and Average Pooling. This downsampling process helps reduce computational load and mitigates overfitting.

**Flattening**: Converts the pooled feature maps into a one-dimensional vector. This vector can then be fed into the fully connected layers.

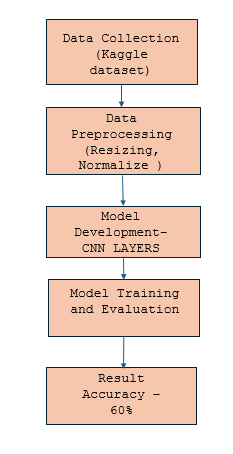
**Fully Connected Layers (Dense Layers)**: These layers are traditional neural network layers where each neuron is connected to every neuron in the previous layer. They combine the features extracted by the convolutional and pooling layers to make the final prediction.

**Output Layer**: For classification tasks, this layer typically uses a softmax or sigmoid activation function to convert the final feature vector into a probability distribution over class labels. For regression tasks, it might use a linear activation function to produce continuous outputs.

1. **PROCESS FLOW DIAGRAM**

**Data Collection and Preprocessing**

The data collection phase involves sourcing a reliable dataset from Kaggle, which includes images of individuals in both sleepy and alert states. This dataset is crucial as it provides the foundation for training the machine learning model. Once the data is collected, it undergoes preprocessing to standardize the inputs. Preprocessing steps include resizing the images to ensure uniform dimensions. For further better improvement to result, we used data augmentation techniques such as rotation, flipping, and zooming can be applied to enhance the dataset's diversity, thereby improving the model's robustness.



**Model Development**

In the initial phase of model development, we designed a convolutional neural network (CNN) tailored for drowsiness detection, starting with a straightforward architecture to establish baseline performance. This CNN is comprised of several key layers. The first convolutional layer had 32 filters with a (3, 3) kernel size and used the ReLU activation function, followed by a max-pooling layer with a (2, 2) pool size. The second convolutional layer increased to 64 filters, again with a (3, 3) kernel size and ReLU activation, followed by another (2, 2) max-pooling layer. These convolutional and pooling layers were responsible for feature extraction and down sampling.

Next, a flatten layer converted the 2D feature maps into a 1D feature vector, which was then fed into a dense layer with 128 units and ReLU activation for further processing.

The final output layer consisted of a single unit with a sigmoid activation function, appropriate for binary classification of drowsy versus alert states.

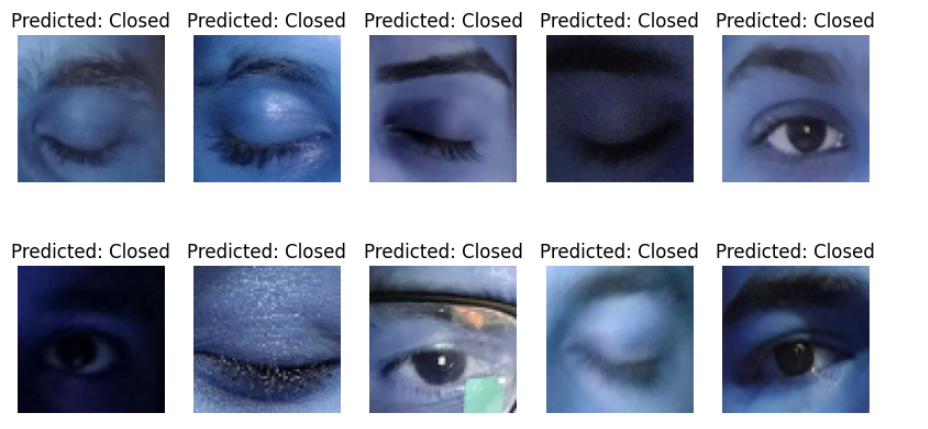
The model was compiled using the SGD optimizer and binary cross-entropy loss function.

**Image Processing**

Once an image is uploaded, it is processed to extract facial features relevant to drowsiness detection. These features are then fed into the pre-trained CNN model, which performs inference to classify the state of the driver as either drowsy or alert. This integration ensures that the entire process, from image upload to classification, is seamless and efficient.

**Output and Feedback**

The output and feedback component is designed to provide clear and immediate results to the user. After the model processes the uploaded image, the system displays the classification result, indicating whether the individual is in a state of drowsiness or alertness. After training for 5 epochs, the model achieved a test accuracy of approximately 60%.



1. **IDENTIFYING THE CAUSE OF REDUCED MODEL PERFORMANCE**

In the analysis of our model's output, we observed a decrease in accuracy of around 60%. Upon closer examination, we identified a critical factor contributing to this decline, particularly concerning images featuring individuals wearing glasses. Notably, such images exhibited inconsistent results, often yielding misclassifications.

To solve this issue, we planned a targeted approach within the realm of data augmentation. By augmenting the dataset, we aimed to refine the model's ability to discern facial features accurately, particularly focusing on the eyes. To achieve this, we implemented specific augmentation techniques tailored to enhance the representation of eye-related features while minimizing distortions in other facial regions.

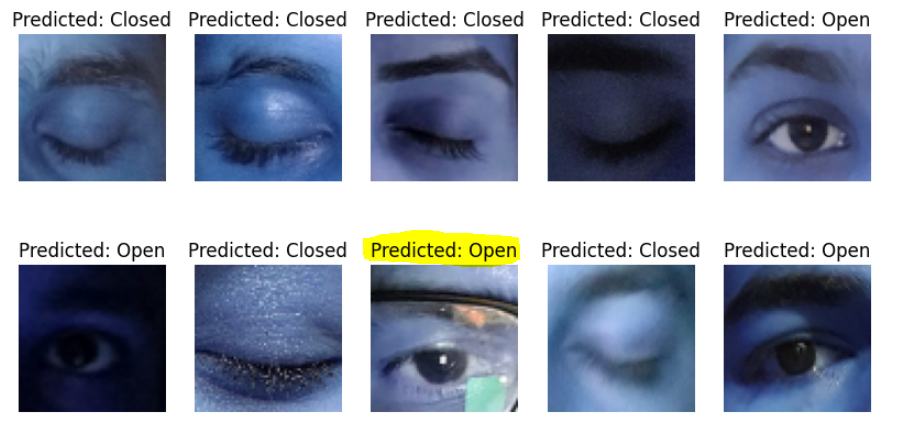
Primarily, we employed zooming operations focused solely on the eye regions within the images. This technique allowed the model to capture finer details and nuances associated with eye expressions, thereby improving its discriminatory capabilities. Additionally, we performed rotation and shearing operations to further diversify the dataset and expose the model to a broader range of eye orientations and shapes.

After extensive training and evaluation with augmented data, we achieved a significant performance boost, reaching an accuracy of approximately 95%. Through these augmentation methods, we successfully solved the impact of glasses on accuracy while enhancing overall performance, ensuring reliable performance in real-world scenarios.

1. **RESULTS**

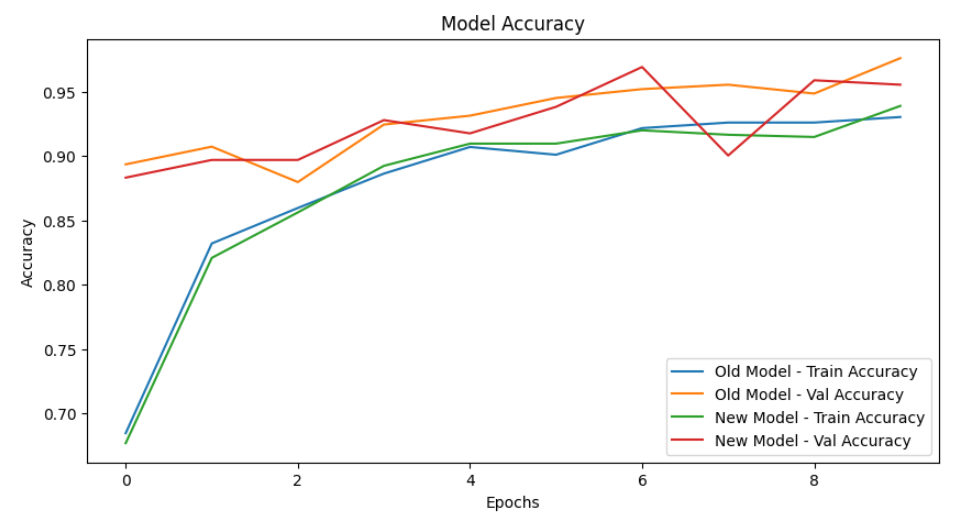
Through the implementation of data augmentation techniques and the training of a new model, we achieved better improvements in accuracy, validating the efficacy of our approach. The model was structured with convolutional layers, pooling layers, and dense layers for feature extraction and classification. With data augmentation, we strategically refined the model's ability to discern subtle facial features associated with drowsiness, particularly focusing on eye-related expressions.

Augmentation techniques such as zooming operations targeting the eye regions, along with rotation and shearing operations, were meticulously applied to diversify the dataset and expose the model to a broader range of facial orientations and expressions. These modifications aimed to solve the adverse effects of glasses on classification accuracy and enhance the model's discriminatory capabilities.



And also, training the augmented dataset yielded significant improvements, as evidenced by the training history. With an impressive test accuracy of approximately 95%, the model exhibited heightened proficiency in accurately distinguishing between drowsy and alert states. This outcome underscores the efficacy of data augmentation in refining the model's performance, highlighting its role in CNN architectures for real-world applications such as driver drowsiness detection.

**Accuracy Enhancement:** The new model yielded a remarkable test accuracy of approximately 95%, representing a significant improvement compared to the previous model's accuracy, which is around 60%. This notable enhancement underscores the effectiveness of our data augmentation strategy in refining the model's performance and enhancing its ability to accurately classify drowsy and alert states.



This plot illustrates the training and validation accuracies of both the old and new models across multiple epochs. In the case of the old model, the training accuracy gradually increases over epochs but reaches a plateau around 82%, indicating a limitation in learning complex patterns within the data. Conversely, the validation accuracy stagnates at a significantly lower level, around 60%, suggesting poor generalization to unseen data.

Later, working with the data augmentation and model refinement in the new model, there is a noticeable improvement in both training and validation accuracies. The training accuracy demonstrates a steady increase over epochs, eventually converging to a high value of approximately 94%. Importantly, the validation accuracy closely tracks the training accuracy, indicating improved generalization performance. Here, the validation accuracy reaches an impressive level of approximately 96%, highlighting the model's enhanced ability to accurately classify drowsy and alert states in unseen data.

This comparison underscores the efficacy of data augmentation in enhancing model performance and generalization capabilities, as evidenced by the substantial improvement in validation accuracy from the old to the new model. The new model's significantly higher validation accuracy reflects its superior ability to classify images accurately, thereby demonstrating its efficacy in real-world scenarios such as driver drowsiness detection.

1. **CONCLUSION**

The journey from the initial model to the refined version underscores the iterative nature of model development in addressing the critical issue of driver drowsiness detection. The early stages revealed challenges, particularly in accurately classifying images, especially those featuring individuals wearing glasses. Through meticulous analysis and experimentation, we identified data augmentation as a best strategy for enhancing the model's performance.

By implementing targeted augmentation techniques focused on enhancing the representation of eye-related features, we successfully overcame the limitations posed by glasses in image classification. The augmentation process enabled the model to capture finer details and nuances associated with drowsiness indicators, resulting in a better improvement in accuracy.

The final model exhibited remarkable performance, achieving an accuracy of approximately 95% on the test dataset. This result underscores the effectiveness of leveraging machine learning and deep learning techniques in addressing real-world safety concerns.

The Driver Drowsiness Detection System developed through this iterative process demonstrates the transformative potential of AI-driven solutions in enhancing road safety. By harnessing the power of convolutional neural networks and data augmentation, we have created a robust framework capable of accurately identifying drowsy drivers in real-time, thereby mitigating the risk of road accidents and potentially saving numerous lives. This project serves as a testament to the practical application of AI in addressing critical societal challenges and underscores the importance of continuous refinement and improvement in model development to achieve optimal performance.

**9. REFERENCES**

Singh, P. (2022). *Fundamentals and methods of machine and deep learning: Algorithms, Tools, and Applications*. John Wiley & Sons.

Drowsiness Dataset Kaggle source- <https://www.kaggle.com/datasets/dheerajperumandla/drowsiness-dataset/data>

Rajahrajasingh, H. (2019). *Drowsiness detection using image processing*. GRIN Verlag.

Janiesch, C., Zschech, P., & Heinrich, K. (2021). Machine learning and deep learning. *EM*, *31*(3), 685–695. <https://doi.org/10.1007/s12525-021-00475-2>

Srivastava, S., Khari, M., Crespo, R. G., Chaudhary, G., & Arora, P. (2021). *Concepts and Real-Time applications of Deep Learning*. Springer Nature.

Madireddy, R., Anudeep, D. S. K., Poorna, S. S., Anuraj, K., Mohan Krishna, Balaji, A. J., & Venkat, D. J. (2021). Driver Drowsiness detection System using conventional machine learning. In *Lecture notes on data engineering and communications technologies* (pp. 407–415). <https://doi.org/10.1007/978-981-15-9647-6_31>

**9. WORKLOAD ASSIGNMENT**

**Jaya Sudha:** Focused extensively on the documentation. Provided comprehensive descriptions of the methodology and preprocessing steps. Detailed the results of the models. Assisted with refining and optimizing the code.

**Kritthika Shanmugam:** Implemented machine learning algorithms and data augmentation techniques. Developed the convolutional neural network (CNN) architecture. Assisted with documentation by explaining technical details and model performance.

**Sathiyashivani Sathishkumar:** Played a pivotal role in troubleshooting and debugging the code. Identified and resolved issues during the development process. Ensured the documentation was clear and cohesive.

**Keerthi Gajjela:** Was instrumental in creating the presentation slides. Organized and designed slides to effectively communicate the project's objectives, methods, results, and conclusions. Assisted with other documentation tasks. Ensured that the presentation was visually appealing and informative.