CNN Model for Driver Drowsiness Detection

Poorna Chand Addala
Department of Information Systems (MS in Business Analytics and Information Systems)
University of South Florida
Tampa, USA
poornachand@usf.edu
(U44711817)

Rupesh Purum
Department of Information Systems (MS in Business Analytics and Information Systems)
University of South Florida
Tampa, USA
rupeshpurum@usf.edu
(U17641787)

Vamsi Krishna Ala
Department of Information Systems (MS in Business Analytics and Information Systems)
University of South Florida
Tampa, USA
av367@usf.edu
(U91394366)

Rohit Vadlala
Department of Information Systems (MS in Business Analytics and Information Systems)
University of South Florida
Tampa, USA
rohitvadlala@usf.edu
(U93298968)

***Abstract*—This study addresses the significant public health issue posed by driver drowsiness, a major contributor to motor vehicle crashes. A convolutional neural network (CNN) model is developed using the Driver Drowsiness Dataset (DDD) to detect drowsiness through facial indicators. The model's performance is fine-tuned through hyperparameter optimization and validated in real-world scenarios with the aim of reducing drowsy driving incidents, thereby enhancing road safety.**

Methods:

The project employs a deep learning approach using a CNN, tailored specifically for facial recognition tasks to detect driver drowsiness. Over 41,790 RGB facial images from the DDD are utilized, classified into 'Drowsy' and 'Non-Drowsy'. Preprocessing steps include image normalization and data augmentation techniques like rotation, scaling, and lighting adjustments to enhance model robustness. The CNN architecture involves multiple convolutional and pooling layers, fully connected layers, and dropout layers to prevent overfitting. Hyperparameters such as learning rate, batch size, and number of epochs are meticulously optimized.

Results:

The CNN model demonstrates high accuracy in distinguishing between 'Drowsy' and 'Non-Drowsy' states from facial images. Through rigorous training and validation, the optimized model achieves robust performance across varied lighting and environmental conditions, essential for real-time processing in diverse driving scenarios. The validation results confirm the model's effectiveness in real-world applications, with significant improvements in detection speeds and reliability.

Conclusion:

The developed CNN model effectively identifies driver drowsiness using facial cues, offering a promising tool for enhancing road safety. The successful application of the model in detecting drowsy states underscores the potential of deep learning technologies in critical safety applications. Future work will focus on integrating the model into real-time monitoring systems in vehicles and further refining its performance across broader demographic variations to ensure universal applicability and reliability.

***Keywords*—** *Driver Drowsiness Detection, Convolutional Neural Networks, Real-Time Systems, Facial Recognition, Deep Learning, Public Health Safety.*

I. Introduction

Driver drowsiness is an important factor contributing to road accidents, resulting in significant mortality, morbidity, and economic costs annually in the United States. Many of these accidents can be avoided with effective detection and timely intervention. The latest developments in deep learning and facial recognition technologies offer an exciting prospect for developing reliable real-time drowsiness detection systems. The Driver Drowsiness Dataset (DDD), which contains over 41,790 RGB images classified into 'Drowsy' and 'Non-Drowsy' states, offers a robust foundation for training a CNN model. The goal of this research is to use this dataset to create a deep-learning model that can recognize driver drowsiness with accuracy. The model's efficiency can be fine-tuned and validated to guarantee that it meets the high reliability that is required for real-world applications, ultimately contributing to more secure driving environments.

II. Problem Identification For Driver Drowsiness Detection Using CNN Model

A. Introduction

First, confirm Driver drowsiness is an essential safety issue that contributes especially to road accidents globally. Driver drowsiness-related accidents have become a serious public health concern as a result of longer commutes and increased traffic density. The issue of driver drowsiness detection is identified in this paper, highlighting the need for a dependable, real-time solution based on cutting-edge deep-learning techniques.

B. Defining the problem

The likelihood of accidents is greatly increased when drivers who are drowsy tend to have poorer attention spans, slower reaction times, and impaired decision-making. Data show that a significant portion of both fatal and non-fatal traffic accidents are directly related to drowsiness: the American Automobile Association (AAA) estimates that fatigue is a factor in 21% of fatal fatalities. Driver fatigue is a contributing factor in about

Identify the applicable funding agency here. If none, delete this text box.

20% of all motor vehicle incidents in the United States, resulting in over 300,000 crashes that are reported to the police and up to 6,400 fatalities per year. The main difficulty is identifying drowsiness quickly and accurately enough to stop accidents before they happen. Conventional techniques, such as tracking eyelid closures or keeping an eye on steering wheel movements, can be unpleasant and have accuracy issues.

C. Role of Deep Learning

New developments in deep learning have opened the door for innovative approaches to monitoring and predicting driver states. Particularly Convolutional Neural Networks (CNNs) have demonstrated potential in identifying intricate patterns and characteristics in image data, which makes them ideal for facial recognition applications including the identification of drowsiness features.

D. Utilization of the Driver Drowsiness Dataset (DDD)

The Driver Drowsiness Dataset (DDD) offers a robust foundation for developing a CNN model. It consists of:

- Over 41,790 RGB facial images, labelled as 'Drowsy' or 'Non-Drowsy'.
- Images processed with the Viola-Jones algorithm, focusing on facial recognition and state detection.

This dataset enables the training of a CNN to detect subtle facial cues indicative of drowsiness, such as drooping eyelids, yawning, and head tilting.

E. Challenges in Developing a CNN Model

While CNNs offer a powerful tool for this application, several challenges need addressing:

Variability in Lighting and Environmental Conditions: The model needs to function consistently across a range of lighting scenarios and inside diverse vehicle interiors.

Real-time Processing: High computational efficiency is necessary for the model to assess and deliver feedback in real-time, which is necessary for the solution to be practical.

Generalization: The model ought to exhibit good generalization abilities without any bias across various demographic groups and individual facial features.

F. Objectives of the CNN Model development

The project aims to develop a CNN-based model that can:

• Accurately differentiate between 'Drowsy' and 'Non-Drowsy' states from facial images.

• Enhance performance by adjusting hyperparameters including batch size, regularization techniques, and learning rate.

• Validate the accuracy and reliability of the model by using it in real-world scenarios.

G. Conclusion and Future Directions

The identification of driver drowsiness is a serious challenge as well as an opportunity to apply cutting-edge AI techniques in practical, life-saving applications. The project aims to improve road safety by offering a tool for early diagnosis of driver fatigue, thereby lowering the chance of drowsy driving accidents. It does this by utilizing a CNN model trained on the DDD. Further research could examine the system's ability to be implemented in real-time scenarios and test it in a variety of environmental settings to make sure it is reliable and universally applicable

III. Data Preparation for Driver Drowsiness Detection Using CNN Model

The development of a Convolutional Neural Network (CNN) model for detecting driver drowsiness encompasses important data preparation stages that would influence the effectiveness and reliability of the final predictive model. The objective of this report is to provide great detail about the data preparation processes implemented for the project working with the Driver Drowsiness Dataset (DDD). Effective data preparation is required for training robust deep learning models that are able to operate under various real-world scenarios.

A. Dataset acquisition and characteristics

The Driver Drowsiness Dataset (DDD) is composed predominantly of RGB facial images that are specifically formatted to facilitate facial recognition and state detection tasks. The dataset comprises over 41,790 images divided into two categories: "Drowsy" and "Non-Drowsy." Each image in the dataset has a resolution of 227 x 227 pixels. Utilizing VLC software, these photos were taken from the video and then processed via the Viola-Jones algorithm to identify and separate the face area, guaranteeing that the data was ready for the intended examination.

B. Data Cleaning and Preprocessing

The data goes through a number of preprocessing stages before it can be used for training:

Quality Check and Cleanup: Initial checks are conducted to make sure the dataset is free of any corrupted or unnecessary files. This guarantees that every image utilized is appropriate for processing and enhances the process of learning.

Normalization: To guarantee consistency in lighting and color distribution, every image is normalized. Normalization assists with reducing discrepancies that occur due to varying shooting conditions, thus enabling the model to emphasize structural and content-related features rather than variations in color.

Data Labelling: Each image has a binary label that indicates whether or not the subject is drowsy. This phase plays an important role in supervised learning, as it dictates the target variable against the predictions made by the model that will be evaluated.

C. Data Augmentation

Data augmentation techniques are used to enhance the CNN model's robustness and simulate different real-world scenarios in which the neural network must operate.

Rotation, Scaling, and Flipping: Images are flipped horizontally, resized, and rotated at random. These modifications aid in the model's acquisition of the ability to identify drowsy states across a variety of facial scales and orientations.

Lighting Variation: To simulate various lighting conditions, such as broad sunshine and dimly lit surroundings, which are prevalent in driving scenarios, adjustments are made to the brightness and contrast of the image.

D. Training, Validation and Test split

The dataset is divided into three subsets:

Training Set: A substantial portion of the dataset which is generally around 70%, is utilized for training the model. Here, the model would learn to identify and differentiate states as Drowsy and Non-Drowsy with the help of provided facial images

Validation Set: Approximately 15% of the dataset will be used for validation. During the training phase, this subset is utilized to adjust the model's parameters to avoid overfitting.

Test Set: The test set, which is composed of the remaining 15%, is only used to assess the model's performance following the training phase. This set can be considered as crucial as it evaluates how effectively the model generalizes to new, untested data.

IV. CNN MODEL ARCHITECTURE FOR DRIVER DROWSINESS DETECTION

The CNN model's architectural design, makes use of the Driver Drowsiness Dataset (DDD) to identify and categorize the facial expressions that indicate drowsiness in drivers.

A. Overview

CNNs can automatically learn and optimize the spatial hierarchies of features from image data, which makes them exceptionally effective for image classification applications. The model's purpose is to identify driver drowsiness by distinguishing between "drowsy" and "non-drowsy" states using face features that are taken from input images.

B. Architecture Details

Input Layer-

Image Input: The input layer can handle images that are 227 * 227 pixels in size. Every RGB image in the DDD passes through this layer initially.

Convolutional Layers-

First Convolutional Layer: This layer performs convolution operations using filters (kernels) to capture basic properties like edges and simple textures. A common selection could be 32 3x3 filters.

Activation Function: A Rectified Linear Unit (ReLU) activation function is applied after every convolution operation to add non-linearity and enable the model to learn more intricate patterns.

Pooling Layers-

Max Pooling: Max pooling layers are used to reduce the dimensionality of the image data after convolution, which aids in lowering the number of parameters and the computing burden. In most cases, a 2x2 pooling size is used for this.

Additional Convolutional and Pooling Layers-

The model consists of several deeper convolutional layers, each of which is activated by a ReLU and max pooled. As the depth increases, this stacking aids in the learning of higher-level features.

Fully Connected Layers-

The architecture consists of one or more fully connected layers following a number of convolutional and pooling layers. The dense layers utilise the output from the previous layers that have been flattened in order to perform classification.

Dropout Layers: Dropout layers are inserted amid fully connected layers to prevent overfitting. During training, a predetermined percentage of neurons are randomly dropped by the dropout layers.

Output Layer-

Classification Layer: The final layer, a softmax layer, outputs the probability distributions of the two classes (drowsy and non-drowsy). The class that the model predicts will be the one with the highest probability.

C. Hyperparameter Training

Learning Rate, Batch Size, and Epochs: The effectiveness and efficacy of model training depend on these hyperparameters. The number of epochs dictates how many times the model views the whole dataset, the batch size influences the gradient estimation, and the learning rate controls how quickly the model adapts to the problem.

Regularization Techniques: Fully connected layers could employ strategies like L2 regularization to penalize heavier weights and reduce overfitting.

D. Implementation Considerations

The model is designed to balance computational efficiency and accuracy in order to efficiently identify and categorize states based on facial features. In order to process streamed footage from in-car cameras, the model must be optimized for low latency and high throughput in real-time deployment.

V. TRAINING PROCESS FOR CNN MODEL IN DRIVER DROWSINESS DETECTION

A. Introduction

This section describes in detail how a convolutional neural network (CNN) that uses the Driver sleepiness Dataset (DDD) to identify driver sleepiness is trained. The process of training is crucial for creating a model that can effectively identify, from facial images, whether a driver is alert or drowsy.

B. Preprocessing and Data Augmentation

The dataset is rigorously pre-processed before training in order to enhance and standardize the quality of the data:

Normalization: Normalization ensures that the scale and intensity distribution of each image are consistent, facilitating more effective data-driven model learning.

Augmentation: Augmentation techniques, including rotation (up to 20 degrees), horizontal flipping, and random cropping, are used to expand the dataset's diversity and simulate different real-world scenarios. This strengthens the model's resistance to overfitting and expands its capacity for generalizing new data.

C. Training, Validation and Testing Splits

The dataset is meticulously divided as follows:

The model is trained on a 70% training set, which gives it the ability to analyze an extensive range of scenarios.

A platform for fine-tuning the model's parameters and architecture without influencing its performance is offered by the validation set (15%).

In order to make sure that the learnt model generalizes well to new data, the testing set (15%) is used to assess the model's performance following the training phase.

D. Training Procedure

Hyperparameters

Optimizer: A gradient descent algorithm such as Adam or SGD (with momentum) is preferred over other algorithms, because of its efficiency in converging to the global minimum.

Learning Rate: Normally commences from a larger value (e.g., 0.001) and is gradually reduced depending on the validation loss in order to fine-tune the model training

Batch Size: Selected in accordance with available processing resources, smaller batches frequently offer more robust convergence at the expense of longer computation times.

Regularization

Dropout: In order to prevent the model from memorizing the noise in the training data, randomly dropping units and their connections during training are applied after fully connected layers.

Batch Normalization: It is applied after each convolutional operation which aims to normalize the activations of the previous layer, ultimately facilitating in speeding up the training process and stabilizing the learning.

Epochs and Iterations

Until the validation accuracy plateaus or begins to decline, suggesting possible overfitting, the model is trained for an adequate number of epochs.

E. Monitoring and Adjusting Training

Performance parameters like accuracy and loss are tracked for both training and validation sets during the training phase. Based on these findings, the training process is constantly modified to enhance performance. Typical instances of these modifications include changing the learning rate or the model architecture's complexity.

F. Model Evaluation and Fine Tuning

The model's performance is assessed using preset measures, including accuracy, precision, recall, and F1-score. In order to guide future model or training process changes and fine-tuning, the confusion matrix generated from the test set offers insights into the types of errors the model is producing.

VI. RESULTS

Classification Report:

Precision: Class 0 (Not Drowsy): The precision is 0.54, meaning that 54% of the instances predicted as not drowsy were actually not drowsy. Class 1 (Drowsy): The precision is 0.47, indicating that 47% of the instances predicted as drowsy were actually drowsy.

Recall: Class 0: The recall is 0.87, suggesting that the model correctly identified 87% of the actual not-drowsy instances.

Class 1: The recall is much lower at 0.13, meaning the model only correctly identified 13% of the actual drowsy instances.

F1-score: Class 0: The F1-score is 0.66, reflecting a better balance between precision and recall for predicting not-drowsy instances compared to drowsy ones. Class 1: The F1-score is significantly lower at 0.21, indicating a poor balance between precision and recall for predictions of drowsy instances.

Support: The number of actual occurrences of each class in the specified dataset shows 4469 instances for class 0 and 3889 for class 1.

Accuracy: The overall accuracy of the model is 0.53, meaning that 53% of the predictions made by the model are correct.

Macro Average: The macro average for precision, recall, and F1-score across both classes is 0.50, 0.50, and 0.43 respectively, reflecting an average performance that is not very high.

Weighted Average: The weighted average for precision, recall, and F1-score, considering the support for each class, is 0.50, 0.53, and 0.45 respectively.

This classification report indicates that while the model performs significantly better at identifying not-drowsy instances (class 0) compared to drowsy ones (class 1), overall effectiveness is moderate at best. The notable disparity in recall between the two classes highlights a critical weakness in detecting drowsy instances, which are significantly under-identified. This could be particularly concerning in practical scenarios where failing to detect drowsiness could lead to accidents or safety issues.

The precision values indicate that nearly half of the predictions for both classes are incorrect, which coupled with the overall accuracy and macro averages, suggests that the model's current setup might not be adequately tuned or capable of distinguishing effectively between the classes in more ambiguous cases.

VII. CONCLUSION

The current evaluation of the model reveals a stark performance disparity between the two classes, with a significant tendency towards accurately predicting non-drowsy instances (class 0) while severely underperforming in detecting drowsy instances (class 1). This issue is highlighted by the model's low recall for class 1, suggesting it fails to identify a substantial number of crucial positive instances, which could have serious practical implications depending on the application.

Given the model's current limitations, primarily its short training period of only 5 epochs and the potential bias toward the majority class, several steps are recommended to enhance its overall performance and reliability. Firstly, increasing the training duration is advised as it provides the model with more opportunities to learn from the data, potentially improving its ability to generalize and better capture the characteristics of the underrepresented class 1. This could directly improve both precision and recall for class 1, making the model more effective in real-world scenarios.

Moreover, implementing class balancing techniques such as SMOTE, adjusting class weights, or undersampling the majority class could correct the model's tendency to favour class 0. Additionally, experimenting with different model architectures, learning rates, and comprehensive feature engineering could further refine the model's predictive capabilities.

While extending the training epochs, it is crucial to remain vigilant against overfitting by employing strategies like cross-validation and early stopping. These measures will ensure that while the model's familiarity with the training data deepens, its ability to perform well on unseen data is not compromised.

In conclusion, by addressing the noted deficiencies through extended training, class balancing, and methodical adjustments to the model's architecture and features, there is a strong potential to achieve a more balanced and effective model. Continued exploration and experimentation will be vital in moving towards a model that performs robustly across both classes, thereby enhancing its applicability and reliability in practical scenarios where accurate detection of drowsiness is critical.

Acknowledgement

This document is our work which is documented for the project report purpose and aims to look at the potential solutions for drowsiness detection by using the CNN model.

References

[1] Liu, P.; Guo, J.M.; Tseng, S.H.; Wong, K.; Lee, J.D.; Yao, C.C.; Zhu, D. Ocular Recognition for Blinking Eyes. IEEE Trans. Image Process. 2017, 26, 5070–5081.

[2] Pandey, N.; Muppalaneni, N. A novel drowsiness detection model using composite features of head, eye, and facial expression. Neural Comput. Appl. 2022, 34, 13883–13893.

[3] Tibrewal, M.; Srivastava, A.; Kayalvizhi, R. A deep learning approach to detect driver drowsiness. Int. J. Eng. Res. Technol. 2021, 10, 183–189.