Real-time Driver Drowsiness Detection Using Deep Neural Networks

Daniel Rafik Halim
Faculty of Engineering Cairo University, Cairo, Egypt

Mariam Khaled Hanafy
Faculty of Engineering Cairo University, Cairo, Egypt

Youssef Saad Lotfy
Faculty of Engineering Cairo University, Cairo, Egypt

Mohaned Deif
Department of Artificial intelligence, College of Information Technology, Misr University for Science & Technology

Rania Elgohary
Department of Artificial intelligence, College of Information Technology, Misr University for Science & Technology

***Abstract*—Abstract: This paper presents a driver drowsiness detection for accident prevention which is based on the curvature of the eye. Our attempt is to develop a deep learning model that can use the input from a camera in real time by extracting the eyes to detect the drowsiness of the drivers.This paper helps to resolve the problem of drowsiness detection with an accuracy of 96% for test and 99% for validation.**

I. Introduction and Problem Statement

Most fatalities and injuries among humans are caused by road accidents. According to the World Health Organization, injuries from road accidents claim one million lives annually worldwide. A driver puts themselves and other road users in danger when they nod off while operating a vehicle due to lack of sleep, fatigue, or other factors. According to studies on auto accidents, driving when sleepy is a major contributing factor to major auto accidents. These days, it is noted that the primary cause of drowsiness while driving is fatigue. Sleepiness is now the primary factor contributing to the rise in traffic accidents. In fact Numerous fatalities have resulted from drivers who were too tired to drive. Heavy trucks are among the many vehicles that are operated at night. Long-haul drivers have been proven to be more vulnerable to these kinds of incidents. Systems for detecting and monitoring drowsiness were developed in order to prevent it. Most of these models use one of the three techniques to detect the drowsiness levels, them being the physiological methods, behavioural methods, and vehicular based techniques. Some hybrid techniques which combine more than one technique have also been suggested to increase the accuracy of the prediction.

II. Related Work

In a tender, a number of techniques were employed to increase the effectiveness and speed of the sleepiness detection process. This section's primary focus is on the techniques and approaches previously employed to recognize drowsiness. The first approach is based on driving patterns, which additionally consider driving styles, road conditions, and characteristics of the vehicle. To ascertain your driving style, compute the amount of steering wheel movement or lane deviation. [1][2]. To keep a car in its lane when driving, one must maintain steady control of the steering wheel. Driver drowsiness was detected 86% of the time by Krajewski et al. [3] using the association between micro adjustments and fatigue. A lane deviation approach can also be used to ascertain the driver's level of fatigue. Here, the car's location in relation to a lane is monitored and analyzed to search for indicators of fatigue [4].

However, the driving patterns-based approaches rely on the type of vehicle, the driver, and the conditions of the road. Physiological detector data, including electrocardiogram (ECG), electroencephalogram (EEG), and electrooculography (EOG) data, are utilized in the alternative category of approaches. EEG signals offer information about the activity of the brain. Three primary indications are utilized to determine a driver's level of fatigue: delta, theta, and nascence signals. Theta and delta signals rise and nascence signals hardly alter in a tired driver. Based on a delicacy rate of more than 90, Mardi et al. [5] claim that this fashion is the most accurate system. The largest disadvantage of this approach, though, is that it is intrusive. Multiple detectors must be linked to the driver, which may not be pleasant. Conversely, non-intrusive bio signal types are far less accurate. All the previous papers were not practical for daily detection, they are either obstructive or depends on too many attributes. The last possible solution is detecting facial features including yawning, face position, and eye blinking, either separately or all together, Danisman et al. [6] monitors only eye blinking to detect drowsiness it detects visual changes in eye locations using a proposed horizontal symmetry feature of the eyes with a 94% accuracy with a 1% false positive rate. This was a great method but unfortunately the dataset was not accessible. Also this was similar to our method. A recurrent and convolutional neural network is proposed by Magan´ et al. [7] and implemented in a fuzzy logic-based system by extracting numeric features from image . Comparable accuracy is attained by both systems: roughly 65% accuracy over training data and 60% accuracy over test data. Which was actually low but the system was similar to what we had in our mind. However, the fuzzy logicbased system is unique in that it achieves a specificity of 93 % (i.e., the proportion of films in which the driver is not drowsy that are accurately identified) and doesn’t give false alerts which was its strength. Using various light sources, Chang et al.[8] chose the proper RGB channel to extract the LF/HF ratio from the HRV of the PPGI. The primary method for judging drowsiness in the suggested drowsiness detection system is the application of an algorithm to determine the percentage of closed eyelids and the sympathetic/parasympathetic nerve balance index. There are 30 drowsy samples and 10 awake samples in the experiment. 88.9% is the sensitivity which is not high if we want to implement real time detection. In another paper An artificial intelligence-based method for sleepiness detection is proposed by Fauzi et al. [9].In this study, a video featuring the driver’s face is captured using a webcam. The driver’s eyes are located in the face region using the Viola-Jones method, which is used for face detection. The technology will determine if the driver’s eye is awake or sleepy after performing some training analysis with accuracy 94 %. This was the closest paper to our method but unfortunately Viola-Jones method has some limitations such as the training process is slow due to the AdaBoost-based feature selection. It works best for frontally positioned faces but may struggle with tilted or side-profile faces which is bond to happen during driving.

III. Dataset and Features

We used a drowsiness detection dataset which classifies based on whether Eyes are Closed or Open from kaggle. The dataset was split into train and validation. We created a dataset for the test using the same conditions as the original dataset. For the train dataset we had 33,557 images for closed/drowsy eye and 34,362 for open/not drowsy eye. For the validation dataset we had 8,389 images for closed/drowsy eye and 8,590 for open/not drowsy eye. For the test dataset we had 1,057 images for closed/drowsy eye and 1,028 for open/not drowsy eye. The dataset was divided into 80For preprocessing this was based on the chosen pre-trained model.our implemented model was based on inceptionv3 model so we used its preprocessing in which the inputs pixel values are scaled between -1 and 1 also all images are resized to 80X80. For data augmentation We used random rotation with factor(-0.05, 0.05) and a fill mode with nearest. In addition to random zoom with height factor(-0.005, 0.005) and width factor(-0.005, 0.005) and fill mode with nearest. Also we used random brightness with factor(-0.1, 0.09) and value range(-1, 1).

A. Resnet50 Model

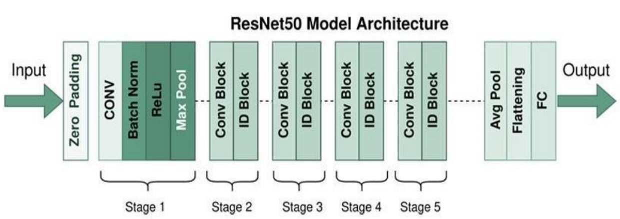


Fig. 7. Resnet50 Architecture

Fig. 1. Train Dataset: Opened

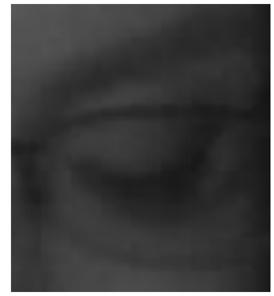


Fig. 2. Train Dataset: Closed

Fig. 3. validation Dataset: Opened

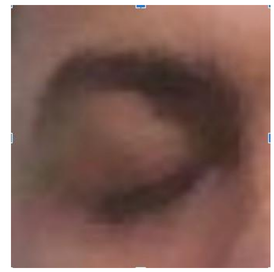


Fig. 4. Validation Dataset: Closed

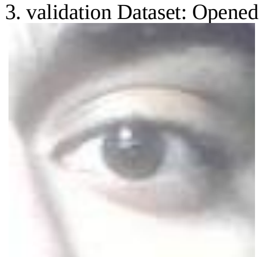


Fig. 5. Testing Dataset: Opened

Fig. 6. Testing Dataset: Closed

IV. Methods:

We used two different architectures at first to make a short comparison based on model behaviour, the two architectures were Inception V3 and ResNet50. While both Inception V3 and ResNet50 are robust deep learning architectures commonly employed for image classification tasks, they diverge in their approaches. Inception V3 emphasises the extraction of multi-scale features utilising inception modules, whereas ResNet50 tackles the challenge of training exceptionally deep networks by incorporating residual connections. Ultimately, our decision to utilise Inception V3 stemmed from its superior accuracy compared to ResNet50. In scenarios where factors such as efficiency in parameter usage, multi-scale feature extraction, computational resource management, adaptability to various input sizes, or achieving state-of-the-art performance on specific tasks are pivotal, Inception V3 emerges as the preferred choice.

B. InecptionV3 Model

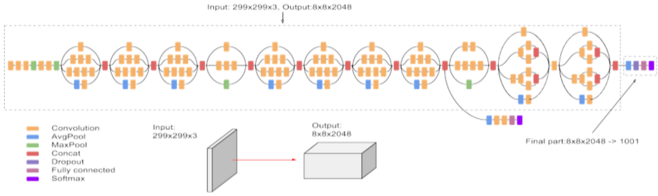


Fig. 8. InceptionV3 Architecture

C. Model Architecture:

We utilized the pre-trained InceptionV3 model as the backbone for our drowsiness detection system. This model was initialized with input shape (80, 80, 3) and pre-trained weights from the ImageNet dataset. To enable fine-tuning of the model on our specific task, all layers of the InceptionV3 model were made trainable.

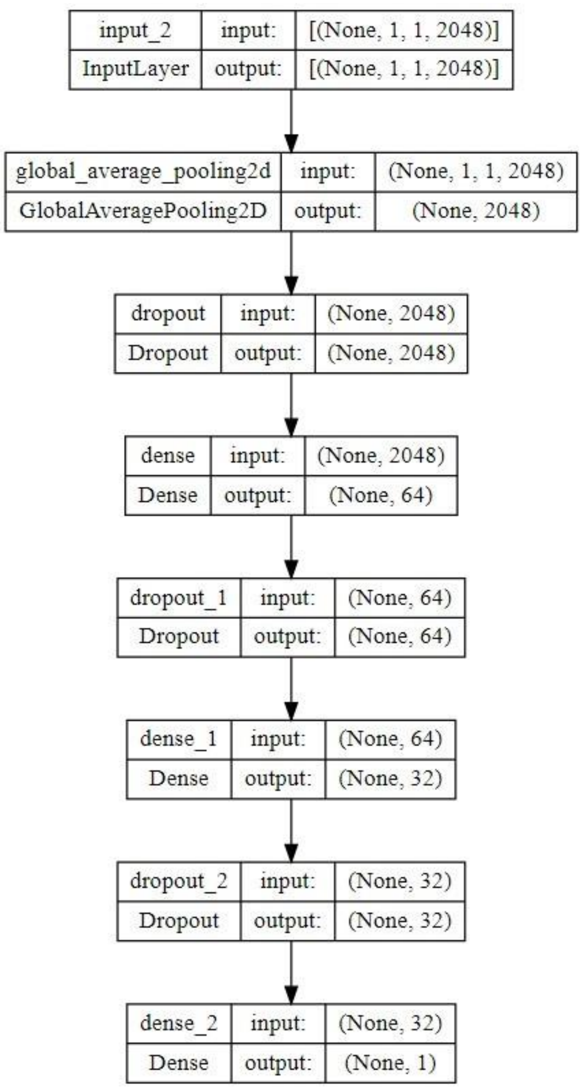


Fig. 9. Model Architecture

D. Custom Layers:

We added custom layers on top of the pre-trained InceptionV3 model to adapt it for drowsiness detection. The global average pooling layer was applied to extract features from the last layer of the InceptionV3 model. Subsequently, dropout layers were introduced to prevent overfitting, with increased dropout rates compared to standard values. Two fully connected dense layers with ReLU activation and L2 regularization were then employed to further process the extracted features. Finally, a sigmoid activation function was utilized in the output layer to produce probabilities indicating the likelihood of drowsiness.

E. Training Setup:

For optimization, we employed the Adam optimizer. The loss function utilized during training was the binary crossentropy loss, which is commonly used for binary classification tasks. Additionally, binary accuracy metrics were employed to evaluate the model's performance during training and validation. The training procedure utilized a batch size determined by the total number of training samples and the desired number of steps per epoch. The loss function utilised during training was the binary cross-entropy loss, defined as:

$$\mathcal{L} = -\frac{1}{N}\sum_{i=1}^{N}\left[y_i \log(p_i) + (1 - y_i)\log(1 - p_i)\right] \quad (1)$$

where:

- L is the binary cross-entropy loss,
- N is the number of samples,
- y_i is the true label for the i-th sample (0 or 1) ,
- p_i is the predicted probability for the i-th sample being in the positive class (output of the sigmoid function).

This comprehensive architecture and training setup were designed to effectively detect drowsiness using eye images, leveraging the capabilities of the InceptionV3 model while customising it for our specific task requirements.

V. Results

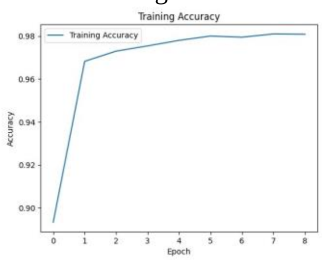
A pre-trained InceptionV3 model is utilised for transfer learning. The model is initialised with weights pre-trained on the ImageNet dataset and its top layer is excluded as typically done in transfer learning scenarios.

Two key hyperparameters are varied to observe their impact on model performance: learning rate and weight decay. Learning rates of 0.0001 and 0.001 are experimented with, as well as weight decay values of 5e-3 and 1e-4. This results in four distinct runs of the model, facilitating a comparative analysis.

Additionally, other hyperparameters are set to further customise the model. Specifically, dropout rates are adjusted to combat overfitting. The dropout rates for the first and second/third dropout layers are set to 0.7 and 0.6, respectively, representing more aggressive regularization strategies compared to typical values. Moreover, a global average pooling layer is employed to reduce the spatial dimensions of the input feature maps. Subsequent dense layers with ReLU activation functions and L2 regularization are added to facilitate feature extraction and reduce the risk of overfitting.

By systematically varying these hyperparameters and observing their effects on model performance, a comprehensive understanding of the model's behavior and its sensitivity to different settings can be gained. This rigorous experimentation and analysis contribute to the robustness and credibility of the research findings.

For Learning Rate: 0.0001 & Weight Decay: 5e-3:



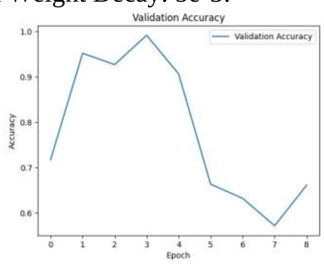
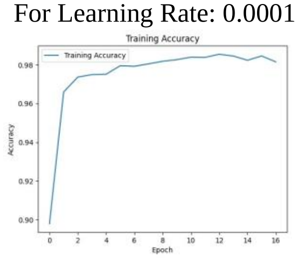


Fig. 10. Training Accuracy for Fig. 11. Validation Accuracy for Learning Rate: 0.0001, Weight De- Learning Rate: 0.0001, Weight Decay: 5e-3, cay: 5e-3

The figures (Fig. 10 and Fig. 11) show the performance of an InceptionV3 model with a learning rate of 0.0001 and weight decay of 5×10^{-3}. The training accuracy improves consistently, while the validation accuracy fluctuates, suggesting potential overfitting.

For Learning Rate: 0.0001 & Weight Decay: 1e-4:



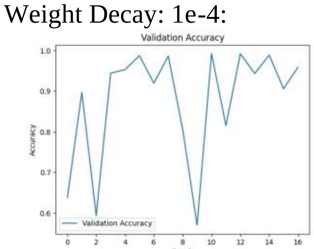
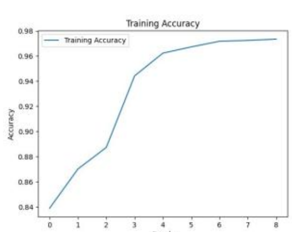


Fig. 12. Training Accuracy for Fig. 13. Validation Accuracy for Learning Rate: 0.0001, Weight De- Learning Rate: 0.0001, Weight Decay: 1e-4 cay: 1 e -4

The second figure (Fig. 12 and Fig. 13) shows the model's performance with a weight decay of 1×10^{-4}. If the validation accuracy graph shows high volatility or a significant drop after a peak, it may indicate overfitting.

For Learning Rate: 0.001 & Weight Decay: 5e-3:



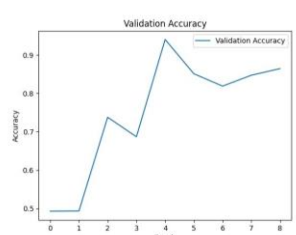
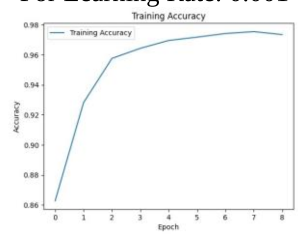


Fig. 14. Training Accuracy for Fig. 15. Validation Accuracy for Learning Rate: 0.001, Weight Decay: Learning Rate: 0.001, Weight Decay:
5e-3 5 e -3

The figures show the performance of an InceptionV3 model with a learning rate of 0.001 and weight decay of 5×10^{-3}. The training accuracy and validation accuracy trends can be observed in Fig. 14 and Fig. 15, respectively.

For Learning Rate: 0.001 & Weight Decay: 1e-4:



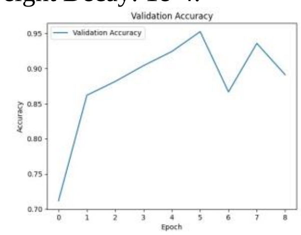


Fig. 16. Training Accuracy for Fig. 17. Validation Accuracy for Learning Rate: 0.001, Weight Decay: Learning Rate: 0.001, Weight Decay:
1e-4 1 e -4

The figures show the performance of an InceptionV3 model with a learning rate of 0.001 and weight decay of 1×10^{-4}. The training accuracy and validation accuracy trends can be observed in Fig. 16 and Fig. 17, respectively.

TABLE I
ACCURACY METRICS FOR DIFFERENT LEARNING RATES AND WEIGHT DECAYS

	Learning Rate: 0.0001	
	Weight Decay: 5e-3	Weight Decay: 1 e -4
Training Accuracy	98.6%	99.2 %
Validation Accuracy	98%	99.1 %
Testing Accuracy	94%	96 %
	Learning Rate: 0.001	
	Weight Decay: 1e-4	Weight Decay: 5 e -3
Training Accuracy	99.2%	97.7 %
Validation Accuracy	99.1%	96.4 %
Testing Accuracy	95%	73 %

$$ACC = \frac{TP+TN}{TP+TN+FP+FN} \quad (2)$$

Accuracy (ACC): This is the proportion of true results (both true positives and true negatives) among the total number of cases examined.

$$PR = \frac{TP}{TP+FP} \quad (3)$$

Precision (PR or PPV - Positive Predictive Value): This is the proportion of true positive results among the total number of positives predicted by the classifier.

$$R = \frac{TP}{TP+FN} \quad (4)$$

Recall (RC or TPR - True Positive Rate): This is the proportion of true positive results among the total number of actual positives.

$$F1 = 2 \times \frac{PR \times R}{PR + R} = 2 \times \frac{TP}{2TP + FP + FN} \quad (5)$$

F1 Score: This is the harmonic mean of Precision and Recall, and it tries to find the balance between precision and recall.

$$SPC = \frac{TN}{TN + FP} \quad (6)$$

Specificity (SPC or TNR - True Negative Rate): This is the proportion of true negative results among the total number of actual negatives.

$$FPR = \frac{FP}{FP + TN} \quad (7)$$

False Positive Rate (FPR): This is the proportion of false positives among the total number of actual negatives.

Before we dive into the results of each model, let's briefly explain what a confusion matrix is. A confusion matrix is used to evaluate the performance of a classification model. The matrix has two rows and two columns, with the rows representing the actual labels and the columns representing the predicted labels. The color code seems to represent the number of instances in each cell, with darker shades indicating higher numbers.

Confusion Matrix For the testing Datasets

A. Resnet50

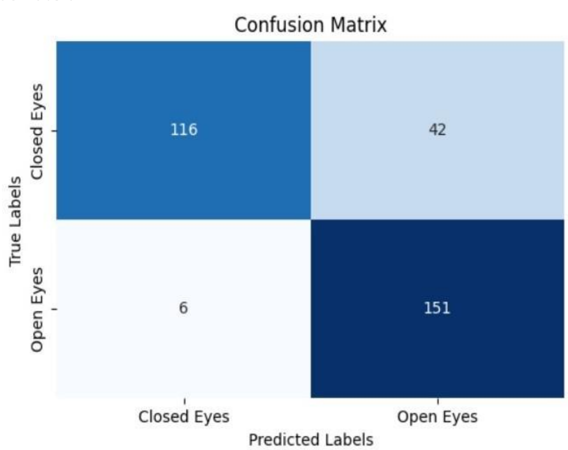


Fig. 18. Confusion matrix for Learning Rate: 0.0001, Weight Decay: 5 e -3

For the ResNet50 model (Figure 18), here's a breakdown of the matrix:

- 'Closed Eyes' predicted as 'Closed Eyes': 116 instances - 'Closed Eyes' predicted as 'Open Eyes': 42 instances - 'Open Eyes' predicted as 'Closed Eyes': 6 instances - 'Open Eyes' predicted as 'Open Eyes': 151 instances

The model was trained with a learning rate of 0.0001 and a weight decay of 5e-3.

B. Inceptionv3

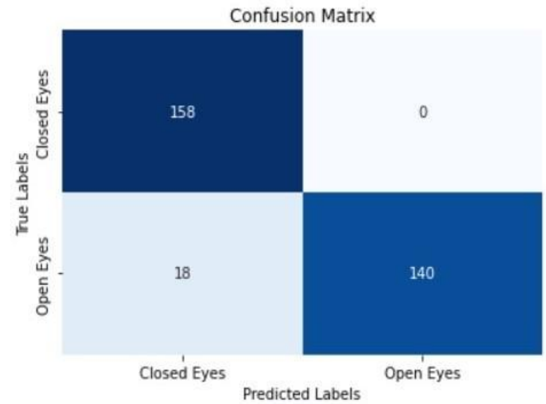


Fig. 19. Confusion matrix for Learning Rate: 0.0001, Weight Decay: 5 e -3

For the Inceptionv3 model (Figure 19), here's a breakdown of the matrix:

'Closed Eyes' predicted as 'Closed Eyes': 158 instances - 'Closed Eyes' predicted as 'Open Eyes': 0 instances - 'Open Eyes' predicted as 'Closed Eyes': 18 instances - 'Open Eyes' predicted as 'Open Eyes': 140 instances.

The model was trained with a learning rate of 0.0001 and a weight decay of 5e-3. As we saw better performance from the inceptionV3 model, we decided to proceed with it.

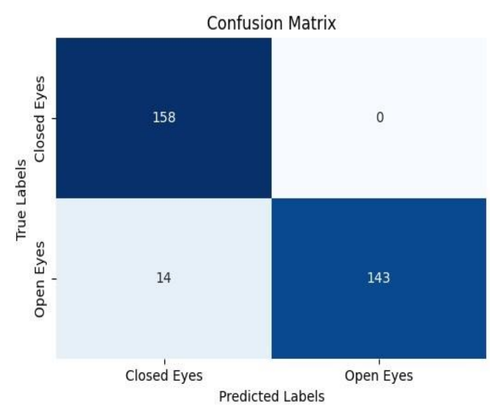


Fig. 20. Confusion matrix for Learning Rate: 0.001, Weight Decay 5 e -3

For the Inceptionv3 model (Figure 20), here's a breakdown of the matrix:

- 'Closed Eyes' predicted as 'Closed Eyes': 158 instances 'Closed Eyes' predicted as 'Open Eyes': 0 instances - 'Open Eyes' predicted as 'Closed Eyes': 14 instances - 'Open Eyes' predicted as 'Open Eyes': 143 instances

The model was trained with a learning rate of 0.001 and a weight decay of 5e-3.

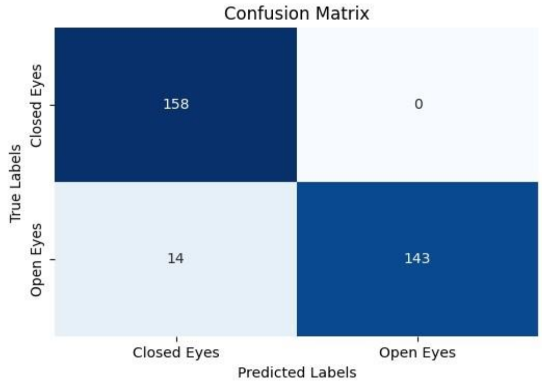


Fig. 21. Confusion matrix for Learning Rate: 0.0001, Weight Decay 1e-4

For the Inceptionv3 model (Figure 21), here's a breakdown of the matrix:

- 'Closed Eyes' predicted as 'Closed Eyes': 158 instances 'Closed Eyes' predicted as 'Open Eyes': 0 instances - 'Open Eyes' predicted as 'Closed Eyes': 14 instances

- 'Open Eyes' predicted as 'Open Eyes': 143 instances

The model was trained with a learning rate of 0.0001 and a weight decay of 1e-4.

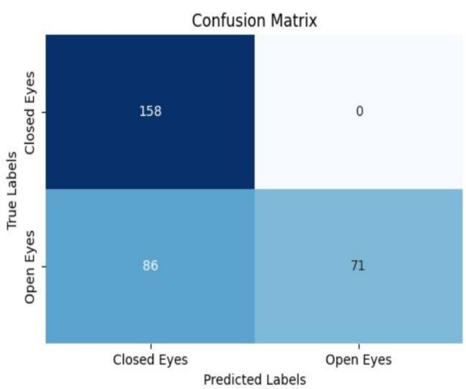


Fig. 22. Confusion matrix for Learning Rate: 0.001, Weight Decay: 1 e -4

For the Inceptionv3 model (Figure 22), here's a breakdown of the matrix:

- 'Closed Eyes' predicted as 'Closed Eyes': 158 instances -
'Closed Eyes' predicted as 'Open Eyes': 0 instances - 'Open Eyes' predicted as 'Closed Eyes': 86 instances - 'Open Eyes' predicted as 'Open Eyes': 71 instances

The model was trained with a learning rate of 0.001 and a weight decay of 5e-3.

VI. Conclusion and Future Work

The paper outlines a thorough investigation into the performance of a pre-trained InceptionV3 model for transfer learning. Two critical hyperparameters, namely learning rate and weight decay, were systematically varied to observe their impact on model performance. Learning rates of 0.0001 and 0.001 were tested alongside weight decay values of 5 e -3 and 1e-4, resulting in four distinct model configurations for comparative analysis.

Additionally, other hyperparameters such as dropout rates were fine-tuned to combat overfitting, with aggressive values of 0.7 and 0.6 set for the first and subsequent dropout layers, respectively. The model architecture included a global average pooling layer followed by dense layers with ReLU activation functions and L2 regularization to facilitate feature extraction and mitigate overfitting risks.

The experimentation yielded insightful results, showcasing the model's sensitivity to different hyperparameter settings. Among the configurations tested, certain combinations demonstrated superior performance over others. For instance, a learning rate of 0.0001 paired with a weight decay of 1e4 achieved the highest accuracy across training, validation, and testing datasets, indicating its effectiveness in learning discriminative features while mitigating overfitting.

The summary table further highlights the performance metrics of each model configuration, with notable disparities in accuracy observed across different parameter settings. Notably, the model with a learning rate of 0.001 and weight decay of 1e-4 exhibited lower accuracy on the testing dataset compared to other configurations, suggesting potential issues with generalization or overfitting.

In conclusion, the rigorous experimentation and analysis conducted in this study contribute to a comprehensive understanding of the model's behavior and its sensitivity to hyperparameter tuning. By identifying optimal configurations, researchers can enhance the performance and generalization capabilities of deep learning models for various tasks. For future exploration, we could experiment with various pretrained model architectures such as VGG, or EfficientNet to gauge their effectiveness for the task at hand. Fine-tuning these architectures or employing neural architecture search (NAS) methods could lead to the discovery of optimal models tailored to our specific dataset.

Additionally, integrating attention mechanisms into the model architecture could enhance its interpretability and performance, especially when dealing with large datasets. Attention mechanisms enable the model to focus on relevant regions of input data, potentially improving both accuracy and insight into model decisions.

VII. References

1) Golz M. Steering wheel behavior based estimation of fatigue. The fifth international driving symposium on human factors in driver assessment, training and vehicle design 2009;118-124.
2) K. Fagerberg.Vehicle-based detection of inattentive driving for integration in an adaptive lane departure warning system Drowsiness detection, M.S. thesis, KTH Signals Sensors and Systems, Stockholm, Sweden, 2004 3) V40 Driver Alert Control (DAC) — Volvo Support UK. (2023, August 6).
4) Mardi Z, Ashtiani SN, Mikaili M. EEG-based drowsiness detection for safe driving using chaotic features and statistical tests. Journal of medical signals and sensors 2011;1:130–137.
5) Danisman T, Bilasco IM, Djeraba C, Ihaddadene N. Drowsy driver detection system using eye blink patterns. Machine and Web Intelligence (ICMWI) IEEE 2010;230233.
6) Hariri B, Abtahi S, Shirmohammadi S, Martel L. A yawning measurement method to detect driver drowsiness. Technical Papers. 2012.
7) E. Magan,´ M. P. Sesmero, J. M. Alonso-Weber, and A. Sanchis, “Driver Drowsiness Detection by Applying Deep Learning Techniques to Sequences of Images,” Applied Sciences (Switzerland), vol. 12, no. 3, Feb. 2022, doi: 10.3390 /app 12031145.
8) Hassan, Esraa, et al. ”COVID-19 diagnosis-based deep learning approaches for COVIDx dataset: A preliminary survey.” Artificial Intelligence for Disease Diagnosis and Prognosis in Smart Healthcare (2023): 107.
9) W. Alkishri, A. Abualkishik, and M. Al-Bahri, “Enhanced Image Processing and Fuzzy Logic Approach for Optimizing Driver Drowsiness Detection,” Applied Computational Intelligence and Soft Computing, vol. 2022, 2022 , doi: 10.1155/2022/9551203.

VIII. Contribution

All authors conceived of the presented idea. M.H. collected the dataset, preprocessed the data and experimented with the data augmentation. Y.S. and M.H. Designed the model architecture. Y.S. addressed overfitting issues, assisted D.R. in real-time analysis. D.R. assisted in designing the model architecture, visualized the results and implemented real-time detection. All authors discussed the results and contributed to the final manuscript.