

Driver Drowsiness Detection Using Deep Learning

Aim

This project aims to investigate the effectiveness of combining face detection and eye region analysis for drowsiness detection. A deep learning model, EfficientNetB0, trained on a dataset of labeled images, will classify eye states (open/closed) and yawning (yes/no). The combined predictions from these classifications will be used to provide a robust drowsiness assessment.

Methodology

1. **Dataset Preparation:** A dataset of facial images categorised into four classes—Open, Closed, yawn, and no_yawn—was used for training and evaluation. The dataset was split into training (80%), validation (20%), and test sets. The distribution of classes within these sets was analysed to ensure a balanced representation.
2. **Data Augmentation:** To enhance model robustness and prevent overfitting, data augmentation was performed on the training set using ImageDataGenerator. This included random rotations, shifts, shears, zooms, horizontal flips, and brightness adjustments.
3. **Model Development:** A pre-trained EfficientNetB0 model, initially trained on the ImageNet dataset, was used as the base model. Transfer learning was employed to leverage the pre-trained weights for feature extraction. The final layers of EfficientNetB0 were unfrozen, and a custom classification head, consisting of dense layers with dropout and L2 regularisation, was added to adapt the model for the four-class drowsiness detection task.
4. **Eye Region Cropping and Quality Assessment:** To improve the accuracy of drowsiness detection, an algorithm was developed to automatically detect and crop the eye region from input images. This algorithm uses Haar cascades for initial eye detection and a quality assessment function to select the best-cropped eye region based on sharpness, contrast, and completeness.

5. **Model Training and Validation:** The model was trained using the augmented training data and validated on a separate validation set. The Adam optimiser and categorical cross-entropy loss function with label smoothing were used for optimisation. Callbacks, including early stopping, ReduceLROnPlateau, and ModelCheckpoint, were implemented to monitor performance and prevent overfitting. The model was trained with optimized hyperparameters for the best results with minimum loss and maximum accuracy.
6. **Model Evaluation:** The trained model was evaluated on the held-out test set using metrics such as accuracy, precision, recall, and the F1-score to assess its performance on unseen data. A confusion matrix was generated to visualize the model's classification performance for each class. The combined predictions for both full-face and cropped eye regions were used for the drowsiness state assessment.
7. **Prediction and Interpretation:** For predictions, both the original image and the cropped eye region are preprocessed and fed to the trained model. The model outputs probabilities for each class, and these probabilities are combined to determine a combined confidence score. Based on the predicted class and its confidence, a final determination of "Drowsy" or "Non-Drowsy" is made.

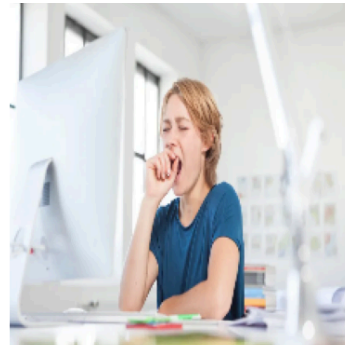
Result

The model achieved an overall accuracy of 100% on the test set. The overall precision and recall were 100% and 100%, respectively. The F1-score, which balances precision and recall, was calculated to be 1.00.

no_yawn and Open (0.92) , State=Non-Drowsy



yawn and Closed (0.86) , State=Drowsy



Class-wise Performance:

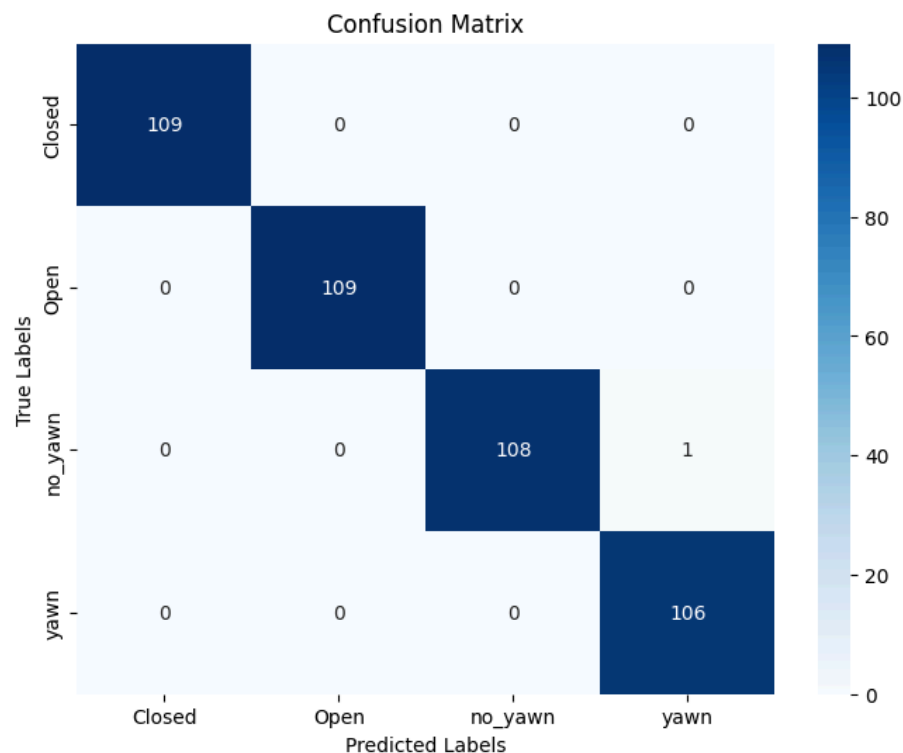
The class-wise performance is detailed in Table 1, presenting the precision and recall for each of the four classes: Open, Closed, yawn, and no_yawn.

Class	Precision (%)	Recall (%)
Open	100	100
Closed	100	100
yawn	99	100
no_yawn	100	99

The class-wise performance shows near-perfect results across all categories. There is only a slight variation in precision for the "yawn" class (99%) and recall for the "no_yawn" class (99%), indicating minimal misclassification between these two categories.

Confusion Matrix:

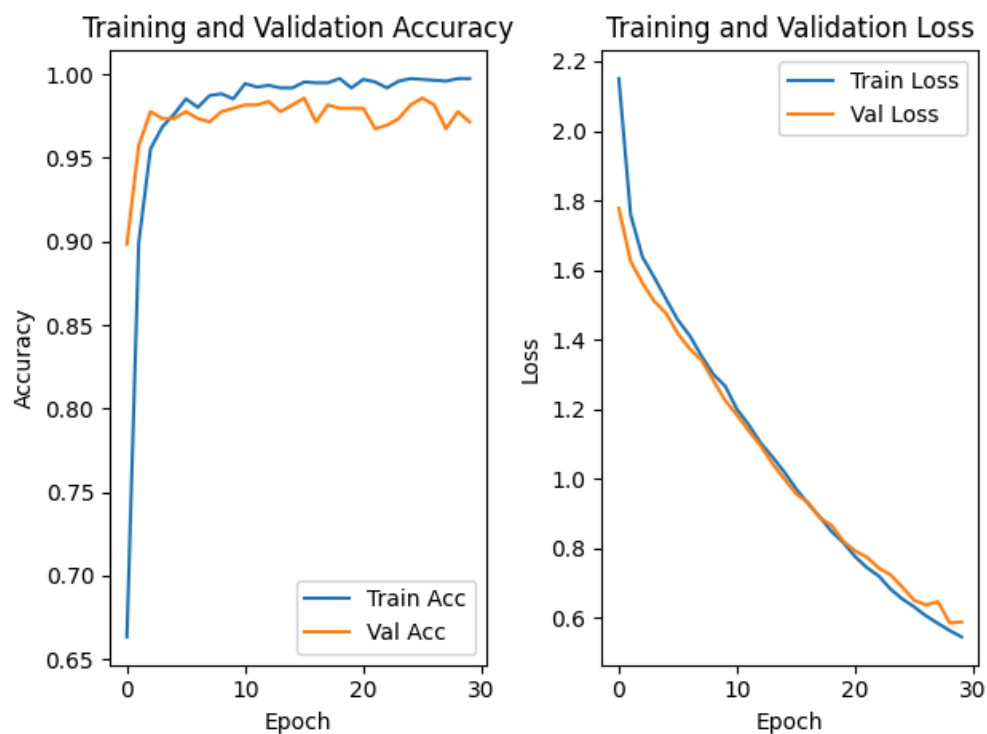
Figure 3 presents the confusion matrix, providing a visual representation of the model's classification performance for each class. The matrix shows the counts of true positive, true negative, false positive, and false negative predictions for each class.



The confusion matrix reveals almost perfect classification performance with only one misclassification: a single instance of a "no_yawn" sample being incorrectly classified as "yawn." All other classes show perfect discrimination with 109 correctly classified samples for both "Closed" and "Open" classes, 108 correctly classified "no_yawn" samples, and 106 correctly classified "yawn" samples.

Accuracy and Loss

The model's performance was evaluated during training using a validation set comprising 20% of the training data. Figure 1 displays the training and validation accuracy and loss curves over 30 epochs.



The training graphs show rapid initial improvement, with both training and validation accuracy quickly reaching above 95% within the first 5 epochs. The training accuracy continued to improve, reaching nearly 100% by epoch 30, while validation accuracy stabilized around 97-98%. Similarly, both training and validation loss decreased consistently throughout training, starting from approximately 2.15 and 1.8, respectively, and reaching around 0.55 and 0.6 by epoch 30. The close alignment between training and validation metrics indicates good generalisation with minimal overfitting, though there are slight signs of overfitting emerging in later epochs.

Reference

<https://www.sciencedirect.com/science/article/pii/S2542660523000288>

<https://www.sciencedirect.com/science/article/pii/S1877050924008846>

Github Link

Telna Chacko [243040] - <https://github.com/telna09/DL-Capstone-Project.git>

Arshida Mahmooda [234110] - https://github.com/arshidamahmooda/DL_Capstone-project

Najiya Fathima [243027] - https://github.com/Najiya-Fathima/Deep-Learning/tree/main/Driver_Drowsiness