Eye Blink Binary Classification

Inuwa Amir Usman

May 25, 2023

1 Introduction

The goal of this research is to show the findings of a convolutional neural network (CNN)-based eye blink detection system trained on the CEW dataset. The system's purpose is to reliably categorize eye pictures as open or closed, indicating whether or not the subject is blinking.

2 Methodology

TensorFlow and the Keras API were used to build the eye blink detection system. Two convolutional layers with ReLU activation, max pooling layers, a flatten layer, and fully connected layers with ReLU activation comprise the CNN architecture. For binary classification, the final output layer employs sigmoid activation.

Data augmentation techniques, such as rescaling, shearing, zooming, and horizontal flipping, were applied to the training set using the `ImageDataGenerator` class. The validation set was only rescaled to ensure consistency.

The Adam optimizer and binary cross-entropy loss were used to construct the model. The training procedure consisted of iterating through the training set for a predetermined number of epochs with a batch size of 32. During training, the validation set was utilized to monitor the model's performance.

3 Results

After training the model, it was evaluated on the validation set. The following results were obtained:

- Validation Loss: 0.1313
- Validation Accuracy: 0.9476
- Precision: 0.9310
- Recall: 0.9687

- F1 Score: 0.9495

The validation accuracy indicates the percentage of correctly classified eye images in the validation set. The precision, recall, and F1 score provide additional insights into the model's performance, particularly in terms of the ability to correctly identify positive and negative examples.

4 Conclusion

The validation accuracy of the eye blink detection system was 94.76 percent. The model's accuracy, recall, and F1 score reflect its ability to properly categorize eye pictures as open or closed. These findings point to the system's potential for real-world applications such as sleepiness monitoring and human-computer interaction.

Further improvements to the system could involve exploring different CNN architectures, hyperparameter tuning, or incorporating more advanced techniques such as recurrent neural networks (RNNs) to capture temporal information.