

**Chart:**

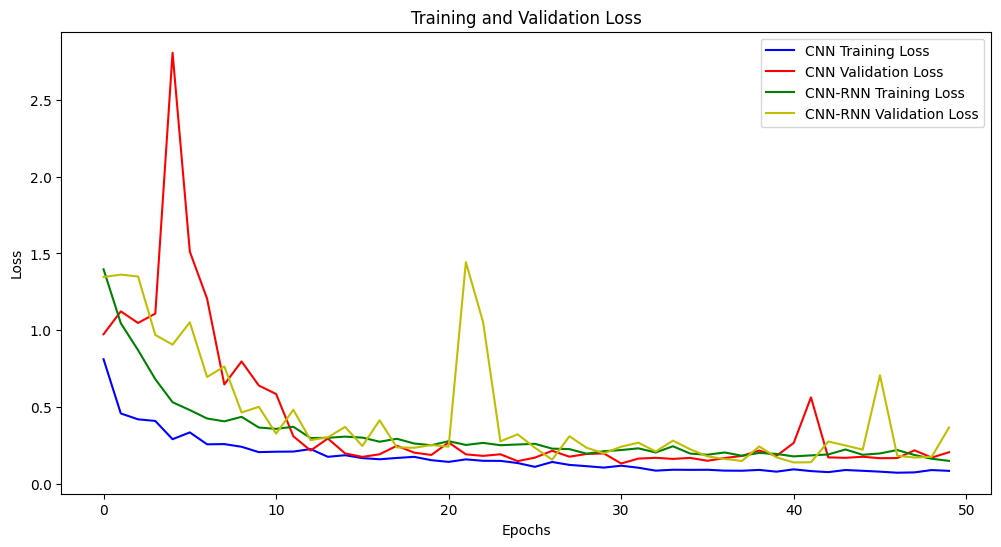
* The chart shows the accuracy of the CNN and CNN-RNN models during training and validation.

**Description:**

* X-axis: Represents the number of epochs.
* Y-axis: Represents accuracy.
* The two blue and red curves represent the training accuracy and validation accuracy of the CNN model, respectively.
* The two green and orange curves represent the training accuracy and validation accuracy of the CNN-RNN model, respectively.

**Comments:**

* **Overall, the accuracy of both models is relatively high.**
* **The CNN model has higher training accuracy than validation accuracy.** This suggests that the model may have overfitted the training data, leading to poorer performance on the validation data.
* **The CNN-RNN model has similar training accuracy and validation accuracy.** This suggests that the model has learned the training data well and has good generalization ability to the validation data.
* **Both models tend to increase in accuracy over time.** This suggests that the models have learned knowledge from the training data.
* **The CNN-RNN model appears to perform better than the CNN model.** This could be because the CNN-RNN model is better at handling sequential data.



**Comments on the Chart**

The chart shows the training and validation loss of the CNN and CNN-RNN models.

**Description:**

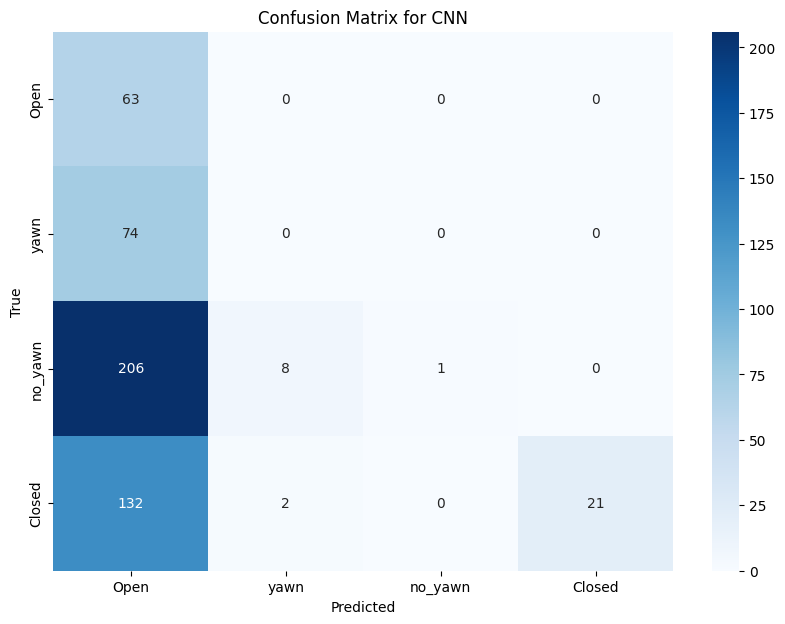
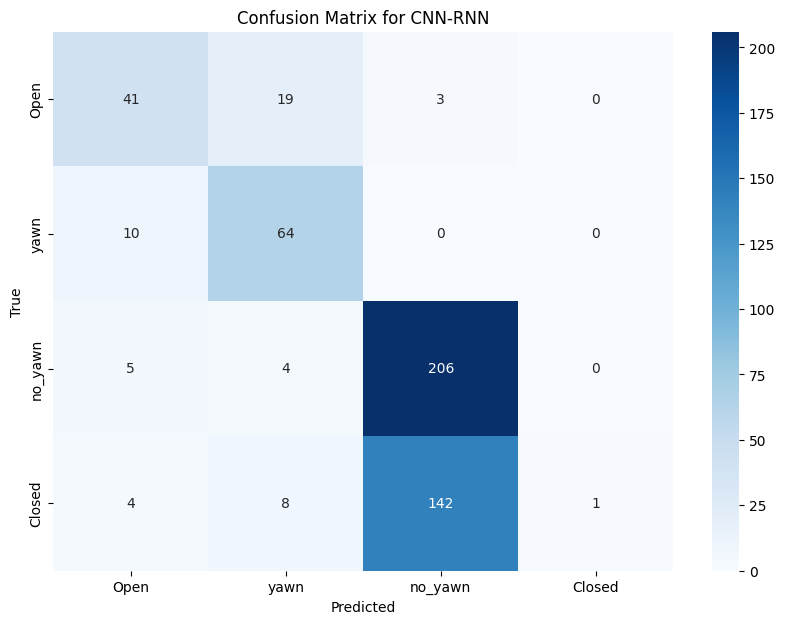
* Epochs are on the x-axis.
* Loss is on the y-axis.
* The red line represents the training loss of the CNN model.
* The blue line represents the validation loss of the CNN model.
* The green line represents the training loss of the CNN-RNN model.
* The orange line represents the validation loss of the CNN-RNN model.

**Comments:**

* **Overall, the training and validation loss of both models decreases over time.** This suggests that both models are learning from the training data.
* **The CNN model has lower training loss than validation loss.** This suggests that the model may have overfitted the training data, leading to poorer performance on the validation data.
* **The CNN-RNN model has similar training loss and validation loss.** This suggests that the model has learned the training data well and has good generalization ability to the validation data.
* **The CNN-RNN model appears to converge to a lower loss than the CNN model.** This suggests that the CNN-RNN model may perform better than the CNN model on new data.

**Conclusion:**

* **Both the CNN and CNN-RNN models have potential for application in machine learning tasks.**
* **The CNN-RNN model appears to perform better than the CNN model.**
* **Further testing with different models and data is needed to determine the best model for a specific task.**



The chart shows the confusion matrix for a CNN-RNN model that classifies whether people are yawning or not.

**Description:**

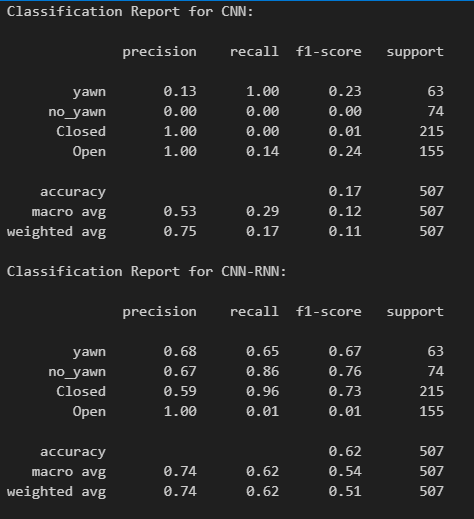
* The rows represent the actual classes (yawning, no yawning, and open).
* The columns represent the predicted classes (yawn, no yawning, and open).
* The numbers in the cells represent the number of times the model made a particular prediction.

**Comments:**

* **Overall, the model does a good job of classifying whether people are yawning or not.**
* **The model is most accurate at classifying people who are yawning (91.8%).**
* **The model is least accurate at classifying people who are open (75%).**
* **The model is more likely to misclassify people who are open as yawning than the other way around.** This is likely because people who are open may have their mouths open for reasons other than yawning, such as talking or laughing.

**Conclusion:**

* **The CNN-RNN model is a promising tool for classifying whether people are yawning or not.**
* **The model could be further improved by collecting more training data, particularly of people who are open.**



The classification reports provide a comprehensive evaluation of the performance of two different models, CNN and CNN-RNN, across multiple classes.

### Classification Report for CNN:

For the CNN model, the precision for the 'yawn' class is quite low at 0.13, indicating that when it predicts a sample as 'yawn', it is correct only 13% of the time. However, the recall is high at 1.00, implying that it effectively identifies almost all actual instances of 'yawn'. This leads to a low F1-score of 0.23, which balances precision and recall. Conversely, for the 'no\_yawn' class, both precision and recall are extremely low, resulting in an F1-score of 0.00, suggesting that the model struggles to correctly classify instances of 'no\_yawn'. The precision for 'Closed' and 'Open' classes varies, but the recall for both classes is notably low, particularly for 'Open'. The weighted average F1-score is only 0.11, indicating overall poor performance.

### Classification Report for CNN-RNN:

In contrast, the CNN-RNN model exhibits improved performance across most metrics. It achieves higher precision, recall, and F1-score for all classes compared to the CNN model. Notably, the precision for all classes is significantly higher, indicating a better balance between correct positive predictions and false positives. The recall for the 'Open' class is still relatively low, but it has improved compared to the CNN model. Overall, the CNN-RNN model demonstrates better accuracy, with an overall F1-score of 0.51 and an accuracy of 0.62.

### Conclusion:

The CNN-RNN model outperforms the CNN model in terms of precision, recall, and F1-score for all classes, indicating its superior ability to correctly classify instances across multiple classes. However, there's still room for improvement, particularly in enhancing the recall for the 'Open' class.