Fall Detection Accuracy Report

August 9, 2023

1 Introduction

This report presents the accuracy assessment of a fall detection program utilizing the Gradient Boosting algorithm. The algorithm predicts fall incidents based on pose keypoints extracted from the MoveNet model for pose estimation. The evaluation is conducted using the UR Fall Detection dataset and the multiple cameras Fall Dataset.

2 Dataset Information

UR Fall Detection Dataset contains 30 falls and 40 confounding activities of daily life sequences. Fall events are recorded by 2 Microsoft Kinect cameras while non-fall events are recorded by only one device and an accelerometer. Multiple Cameras Fall Dataset contains 24 scenarios recorded with 8 video cameras from different perspective in a room. More specific, the first 22 scenarios contain a fall and confounding events, the last 2 ones contain only confounding events.

3 Model Architecture

The Gradient Boosting algorithm, a powerful ensemble method, was chosen for its ability to handle complex relationships in data. The algorithm consists of a sequence of weak learners, typically decision trees, which are combined to form a strong predictive model.

4 Training Details

The Gradient Boosting algorithm was trained using the training subset of the dataset. The learning rate was set to 0.1.

5 Evaluation Metrics

The model's performance was evaluated using the following metrics:

- **Accuracy** is defined as the percentage of correctly predicted output.
- **Recall** is the ability of the classifier to find all the positive instances.
- **Precision** is the measure of how well a classifier does not mis classify instances as positives when they're negatives.
- **F1 Score** is the weighted harmonic mean of precision and recall. The best score for F1 Score is 1.0 and the worst is 0.0.

The performance of the model can also be evaluated visually using the Confusion Matrix and the Reciever Operation Characteristics (ROC) curve.
Confusion matrix is a matrix that summarizes the performance of a machine learning model on a set of test data. It is often used to measure the performance of classification models, which aim to predict a categorical label for each input instance. The matrix displays the number of true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN) produced by the model on the test data.
ROC curve is like a graph that shows how well a classification model performs. It helps us see how the model makes decisions at different levels of certainty. The curve has two lines: one for how often the model correctly identifies positive cases (true positives) and another for how often it mistakenly identifies negative cases as positive (false positives). By looking at this graph, we can understand how good the model is and choose the threshold that gives us the right balance between correct and incorrect predictions. he **Area Under the Curve (AUC)** is the measure of the ability of a binary classifier to distinguish between classes and is used as a summary of the ROC curve. The higher the AUC, the better the model's performance at distinguishing between the positive and negative classes.

6 Results

The results of the model's performance on the testing subset are as follows:

- Accuracy: 86.4%
- Precision: 87.2%
- Recall: 95.8%
- F1-score: 91.3%

6.1 Confusion Matrix

6.2 ROC Curve

Below is the ROC curve for the model:

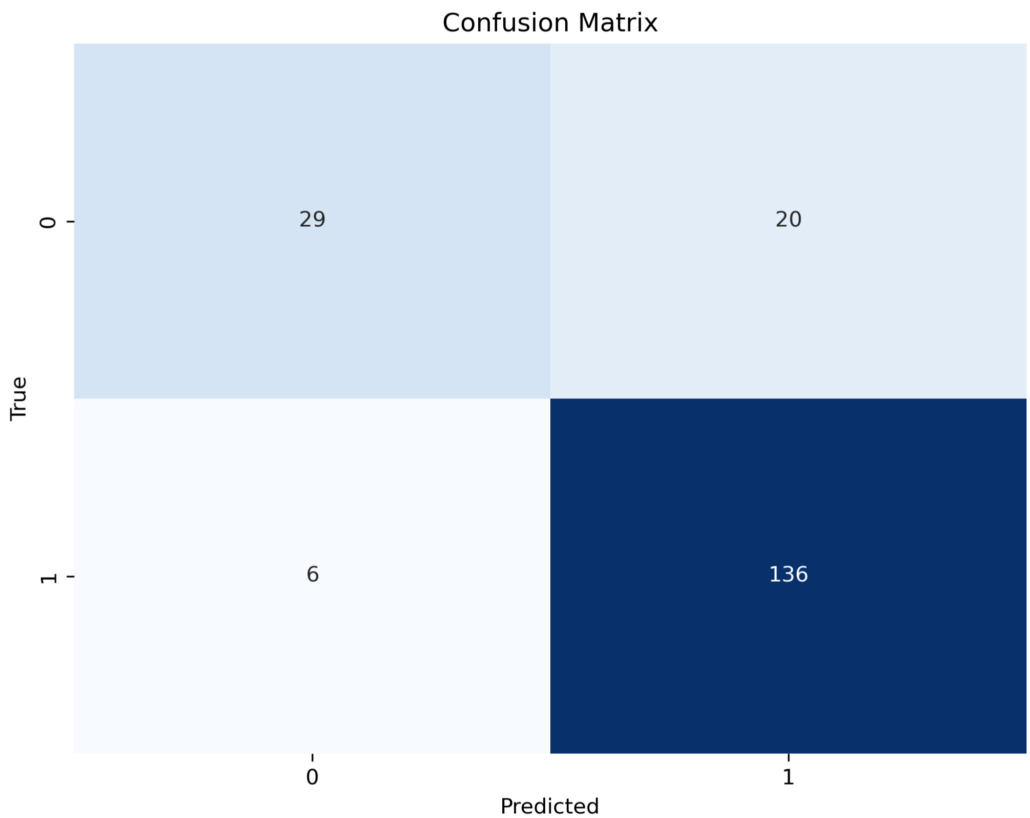

Figure 1: Confusion Matrix

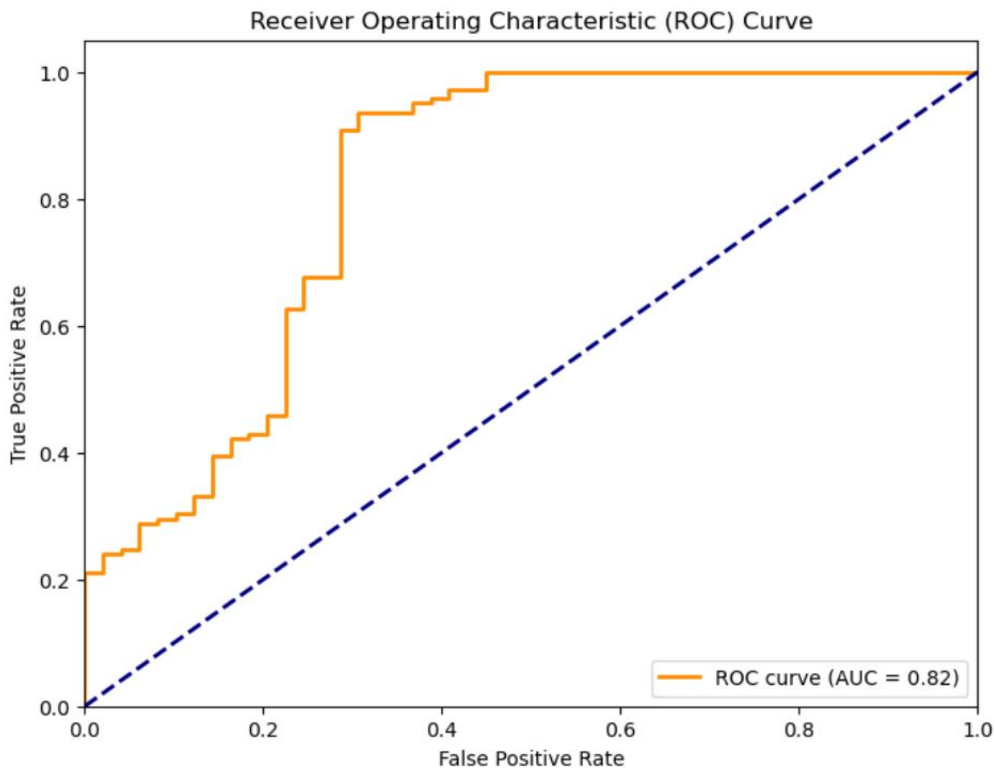

Figure 2: ROC curve

7 Discussion

ROC Curve: The ROC curve plots the true positive rate (TPR) against the false positive rate (FPR) at various threshold settings for a binary classifier. It helps you visualize the trade-off between sensitivity (true positive rate) and specificity (true negative rate) as the discrimination threshold for classifying positive and negative instances changes.

AUC Score: The AUC score is a single scalar value that summarizes the performance of the ROC curve. It represents the area under the ROC curve. AUC scores range from 0 to 1, where:

- AUC = 0.5: The model's performance is similar to random chance.
- AUC $\leq$ 0.5: The model's performance is better than random chance.
- AUC = 1: The model perfectly separates the positive and negative instances.

Interpretation: In our model, we have an AUC score of 0.82. Here's what this score generally implies:

- Good Discrimination: An AUC score of 0.82 suggests that our binary classifier is able to distinguish between positive and negative instances relatively well. It's performing significantly better than random chance.
- Strengths and Limitations: The closer the AUC score is to 1, the better the classifier's performance. However, an AUC of 0.82 is still considered quite good and indicates that our classifier is providing strong discrimination between the two classes.
- Room for Improvement: While an AUC of 0.82 is promising, we might still explore opportunities to improve the model's performance. This could involve fine-tuning hyperparameters, collecting more diverse training data, experimenting with different algorithms.

Overall, an AUC score of 0.82 indicates that our binary classifier is demonstrating good discrimination between positive and negative instances.

Confusion Matrix: The confusion matrix provides valuable insights into the model's performance. In our case, the relatively low number of false negatives (6 instances) is indeed a positive sign. This suggests that the model is doing well at identifying actual falls and is not missing many fall instances. False negatives are critical to minimize, especially in applications like fall detection where missing an actual fall can have serious consequences.

However, the false positives (20 instances) could indicate a potential area for improvement. While the model is generally good at identifying falls, it's making some incorrect predictions of falls when there are none. Depending on the context and the specific application, reducing false positives might be important, especially if false alarms can lead to unnecessary interventions.

Overall, the combination of relatively low false negatives and a good number of true positives suggests that your model is performing well in fall detection, with room for fine-tuning to improve false positives.

8 Future Steps

To further enhance the fall detection program's accuracy, future steps could involve collecting additional labeled data for a more comprehensive dataset. Exploring alternative algorithms and hyperparameter tuning could also lead to performance improvements.

9 Conclusion

In conclusion, the Gradient Boosting algorithm proves to be a valuable tool for fall detection using pose keypoints obtained from the MoveNet model. The model's accuracy, precision, and recall indicate its potential for real-world applications in ensuring prompt responses to fall incidents.