Fall Detection Using Gyroscope Data

Jayani Tripathi

Table of Contents:

Introduction

The fall detection system uses gyroscope data to classify activities and identify potential falls. This system employs a decision tree classification model and derives features from gyroscope signals. It has applications in healthcare, particularly for monitoring the elderly or individuals at risk of falling.

Dataset and Preprocessing

1. **Dataset Structure:**
    - Data files are categorized into directories corresponding to specific activities.
    - Each file contains gyroscope readings (accel_x, accel_y, accel_z) and timestamps.
2. **Preprocessing Steps:**
    - **Magnitude Calculation:** The magnitude of acceleration is calculated: magnitude=x2+y2+z2\text{magnitude} = \sqrt{x^2 + y^2 + z^2} The mean value is subtracted from the magnitude to normalize the data.
    - **Noise Removal:** A low-pass Butterworth filter with a cutoff frequency of 5 Hz is applied to smooth the signal.
3. **Feature Extraction:** Statistical features are extracted for each data window:
    - Mean, maximum, median, minimum, standard deviation, and interquartile range (IQR).
4. **Windowing:** Data is segmented into fixed-sized windows (e.g., 10 seconds), with each window labeled based on the activity.

Model Training

1. **Feature Selection:** The following features were selected:
    - Average (avg), Maximum (max), Median (med), Minimum (min), First Quartile (q25), Third Quartile (q75), Standard Deviation (std).
2. **Classifier:**
    - A Decision Tree Classifier was used with:
        - Criterion: entropy
        - Maximum Depth: 5
3. **Train-Test Split:**
    - Data was split into training (70%) and testing (30%) sets.

4. **Evaluation Metrics:**
    - Accuracy Score
    - Classification Report (Precision, Recall, F1-Score)
    - Confusion Matrix

Live Classification

1. **Input Data:**
    - Live gyroscope readings are processed in real time, containing accel_x, accel_y, and accel_z.
2. **Processing:**
    - Magnitude is calculated, and features are extracted from the live data.
    - A pre-trained Decision Tree model (saved as dt_model.pkl) is used to predict the activity.
3. **Output:**
    - Predicted activity labels are provided in real time.

Results

1. **Model Performance:**
    - The decision tree classifier achieved an accuracy of **X%** on the test set (replace with actual value upon implementation).
    - Classification Report:
2. **Confusion Matrix:** The confusion matrix highlights the classifier's performance, showcasing correct and incorrect predictions across activities.

	precision	recall	f1-score	support
Downstairs	0.60	0.50	0.55	6
Falling	0.83	0.95	0.88	20
Jumping	0.67	0.40	0.50	5
Running	0.75	0.90	0.82	10
Standing	1.00	0.83	0.91	6
Upstairs	0.33	0.60	0.43	5
Walking	1.00	0.67	0.80	15
accuracy			0.76	67
macro avg	0.74	0.69	0.70	67
weighted avg	0.80	0.76	0.76	67

Accuracy on test set: 0.7611940298507462

3. **Live Data Prediction:**
    - The system successfully classified live data with minimal delay.

Conclusion

- The fall detection system demonstrates effective activity classification using gyroscope data and decision tree modeling.
- Robust preprocessing (e.g., noise removal, feature engineering) contributes significantly to its performance.
- Future work may include:
    - Expanding the dataset for diverse activity coverage.
    - Incorporating other classification models like Random Forest or Neural Networks for potential performance improvements.
    - Optimizing parameters for faster real-time predictions.

Note: Please feel free to email me for the code of this project!