Fall Detection System Documentation

21pd39
Varsha S
Psg college of Technology

Introduction

The Fall Detection System aims to address the critical issue of detecting falls among individuals, especially in high-risk areas such as staircases, escalators, and steps. Falls can lead to severe injuries or even be life-threatening, making it crucial to have an efficient and accurate detection mechanism. This document provides an in-depth overview of the system's components, functionality, implementation, and evaluation criteria.

Problem Statement

Falls pose a significant risk, particularly among the elderly and individuals with mobility challenges. The key challenges addressed by the Fall Detection System include:

- Accurate Detection: Ensuring that falls are accurately detected without missing any instances.
- False Positive Reduction: Minimizing false alarms to avoid unnecessary alerts and disruptions.

Technologies Used

The Fall Detection System leverages the following technologies and methodologies:

- YOLO Computer Vision (CV) Library: Used for real-time object detection, including identifying humans in video streams.
- Pose Estimation Algorithm: Extracts key body pose information, aiding in identifying falling motions.
- Open-source Models OWL-ViT and OWL-V2: Integrated to enhance the system's accuracy and robustness.

System Components

The Fall Detection System comprises the following components:

- Input Module: Accepts offline videos in mp4 format for fall detection analysis.
- Object Detection Module: Utilizes YOLO CV Library to identify human subjects in the video frames.
- Pose Estimation Module: Extracts key body pose information to detect falling motions accurately.
- Classification Module: Uses OWL-ViT and OWL-V2 models for enhanced classification accuracy.
- Output Module: Provides visual indicators and alerts for fall detection events.

Implementation Steps

To deploy and utilize the Fall Detection System, follow these steps:

1. Install Required Libraries:

   Install the necessary Python libraries including `cv2`, `cvzone`, `math`, and `ultralytics`.

2. Download Model Files:

   Obtain the YOLO model (`yolov8s.pt`) and the classes file (`classes.txt`) required for object detection.

3. Update File Paths:

   Modify file paths and parameters as necessary in the provided Python script.

4. Run the System:

   Execute the Python script to start the fall detection system, providing it with the desired video input.

Evaluation Criteria

The performance of the Fall Detection System will be evaluated based on the following criteria:

- High Accuracy of Fall Predictions: The system should accurately identify falling motions with minimal false negatives.
- False Positive Minimization: False alarms and unnecessary alerts should be kept to a minimum to maintain system reliability.

Result and Demonstration

The system processes input videos, detects potential falls using a combination of object detection and pose estimation techniques, and generates visual indicators for fall detection events. A demo video showcasing the system's functionality will be provided alongside this documentation.

RESULT - FALL DETECTION

Conclusion

The Fall Detection System addresses a critical need for accurately detecting falls in various environments. By leveraging computer vision techniques and advanced models, it aims to enhance safety and reduce the risks associated with falls, especially in vulnerable populations. Continuous evaluation and improvement will be conducted to ensure the system's effectiveness and reliability.

—----------

This documentation is purely done by me Varsha S for the Yavar internship hackathon.