CSCI 475 Project Milestone: Machine Learning-Based Fall Detection System Using Convolutional Neural Networks

Manish Adhikari
manish.adhikari@bison.howard.edu

Prasun Dhungana
prasun.dhungana@bison.howard.edu

Abstract

Our project aims to develop a machine learning model that uses image frames to detect falls in real time, aiding in faster response and prevention of severe injuries. We leverage the UR Fall Detection (URFD) dataset, containing video clips labeled for fall and non-fall instances, to create a reliable, efficient, and lightweight fall detection system. This system relies on Convolutional Neural Networks (CNNs) to classify frames as "fall" or "no_fall," providing a robust and adaptable model suitable for diverse environments and real-world applications.

1 Introduction/Motivation

Falls among the elderly, whether at home or in care facilities, pose significant health risks, leading to injury and reduced quality of life. Timely detection of falls can improve response times in emergencies and potentially reduce harm. Our goal is to develop a machine learning-based fall detection system that classifies video frames as fall or non-fall instances. By focusing on image frames, this project seeks to create an adaptable and scalable model that can operate in real time with minimal resource demands, making it suitable for edge deployment.

2 Methodology

To achieve an effective fall detection system, our approach involves the following steps:

- **Data Collection and Preprocessing**:
  - **Data Source**: We utilize the URFD dataset, which provides video clips labeled for fall and non-fall events.
  - **Frame Extraction**: We extracted one frame per second from each video, creating a dataset of images classified into two categories: "fall" and "no_fall."
  - **Image Filtering**: For simplicity and efficiency, only RGB images were saved, as depth data is beyond the scope of this project.
- **Exploratory Data Analysis**: We examined the URFD dataset to understand the distribution of fall vs. non-fall instances, assessing the data's characteristics and identifying any potential imbalance.
- **Model Development**: We will implement a Convolutional Neural Network (CNN) architecture to classify images as fall or no_fall. Initial training will establish a baseline performance for accuracy, followed by hyperparameter tuning and further optimization.
- **Baseline CNN Model**: Using a basic CNN architecture, we will train the model on the balanced, preprocessed dataset we are creating from the URFD source and evaluate its performance on accuracy and loss metrics. This baseline model will serve as a foundation for further improvements and experimentation.

3 PROJECT MILESTONES AND PROGRESS UPDATE

- **Data Collection:** Video clips from the UR Fall Detection Data website have been downloaded and are labeled for ”fall” and ”no_fall” events.
- **Frame Extraction:** One frame per second has been extracted from each video file to construct a set of static images. Each frame has been labeled as either ”fall” or ”no_fall”.
- **Image Preprocessing:** The video frames obtained contained RGB and depth field sections spliced side by side. The frames have been split, and only the RGB images have been retained for the dataset for simplicity and efficiency, as per the current scope of the model.
- **Dataset Completion and Model Development:** The image dataset is currently being finalized, with efforts being made to ensure proper labeling and data quality. Once this process is completed, the development of the CNN model for classification will be initiated.

4 CHALLENGES

- **Imbalance in Class Distribution:** Frame extractions from video clips of individuals falling creates images where the individual is stable, unstable, and completely fallen. On the other hand, frames from clips of individuals performing other activities will only have images of individuals not falling. This disparity causes the dataset to have a higher number of ”no_fall” images compared to ”fall” images. This class imbalance could lead to model bias, where the model is more likely to predict the majority class, ”no_fall,” and perform poorly on fall detection.
- **Limited Variability in Dataset:** The images extracted for the dataset were all captured in a similar environment, with consistent lighting and a lack of vibrant colors. This lack of diversity in the dataset can hinder the model's ability to generalize to various real-world settings, where environmental conditions and lighting can vary significantly.
- **Time-Consuming Manual Labeling:** Separating the images into ”fall” and ”no_fall” categories manually has proven to be quite time-consuming.

5 NEXT STEPS

- **Data Balancing:** To tackle the class imbalance, we plan to programmatically remove a subset of ”no_fall” images, ensuring a more balanced dataset. This approach will help reduce model bias towards the majority class, improving the model's ability to detect falls accurately.
- **Data Augmentation:** To address the lack of variability in the images, we will apply data augmentation techniques such as random rotation, color jittering, and the application of various filters. These methods will simulate different lighting conditions, angles, and environments, enhancing the model's robustness and its ability to generalize to real-world scenarios.

6 FINAL GOALS & EVALUATION

Our aim is to achieve a fall detection model with at least 85% accuracy on a validation set, along with high precision and recall scores. Evaluation will focus on assessing the model's robustness across diverse test conditions. To demonstrate usability, we also plan to build a simple interactive UI that allows users to upload images and receive model predictions, adding a practical element to our project.

7 RELATED WORK

A study by Salimi et al. (2022) explores using Deep Neural Networks for human fall detection with pose estimation, utilizing a combination of Time-Distributed Convolutional Long Short-Term Memory (TD-CNN-LSTM) and 1D-CNN models, achieving accuracies of 98% and 97%. Their approach highlights the effectiveness of combining pose estimation with deep learning models for fall detection. While our current project focuses on static image frames without pose estimation, we may consider incorporating techniques similar to Salimi et al. if initial results indicate the need for additional enhancements.

REFERENCES

Bogdan Kwolek and Michal Kepski. Human fall detection on embedded platform using depth maps and wireless accelerometer. *Computer Methods and Programs in Biomedicine*, 117(3):489–501, 2014. ISSN 0169-2607. doi: 10.1016/j.cmpb.2014.09.002.

M Salimi, JJM Machado, and JMRS Tavares. Using deep neural networks for human fall detection based on pose estimation. *Sensors (Basel)*, 22(12):4544, 2022. doi: 10.3390/s22124544.