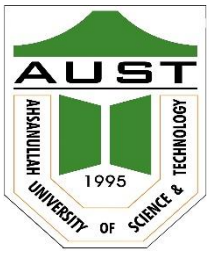

Ahsanullah University of Science and Technology

Department of Computer Science and Engineering

Project Report 02

Course Code - CSE4264

Course Title - Internet of Things Lab

Submitted By -

Lab Group No: 02

Project Group No: 04

Group Members:

Maisha Fahmida Hossain (20200204073)
Sameen Ajreen Rahman (20200204092)
Rehenuma Tabassum (20200204104)
Hasan Farabi (20200204106)

Submitted On -

20.02.2025 (Thursday)

Submitted To -

Md. Reasad Zaman Chowdhury & Ms. Ayesha Banu

FallGuard: Detecting Irregular Human Motion for Elderly Safety

Overview

We are developing a fall detection device primarily for elderly individuals using a Raspberry Pi and a webcam. The system continuously monitors movements using a camera and applies computer vision techniques to detect falls in real time. By training our Machine Learning model on a fall detection dataset, we aim to improve the system's accuracy and reliability. The ultimate goal is to provide a cost-effective and efficient solution that ensures the safety of elderly individuals by detecting falls and triggering alerts.

Progress

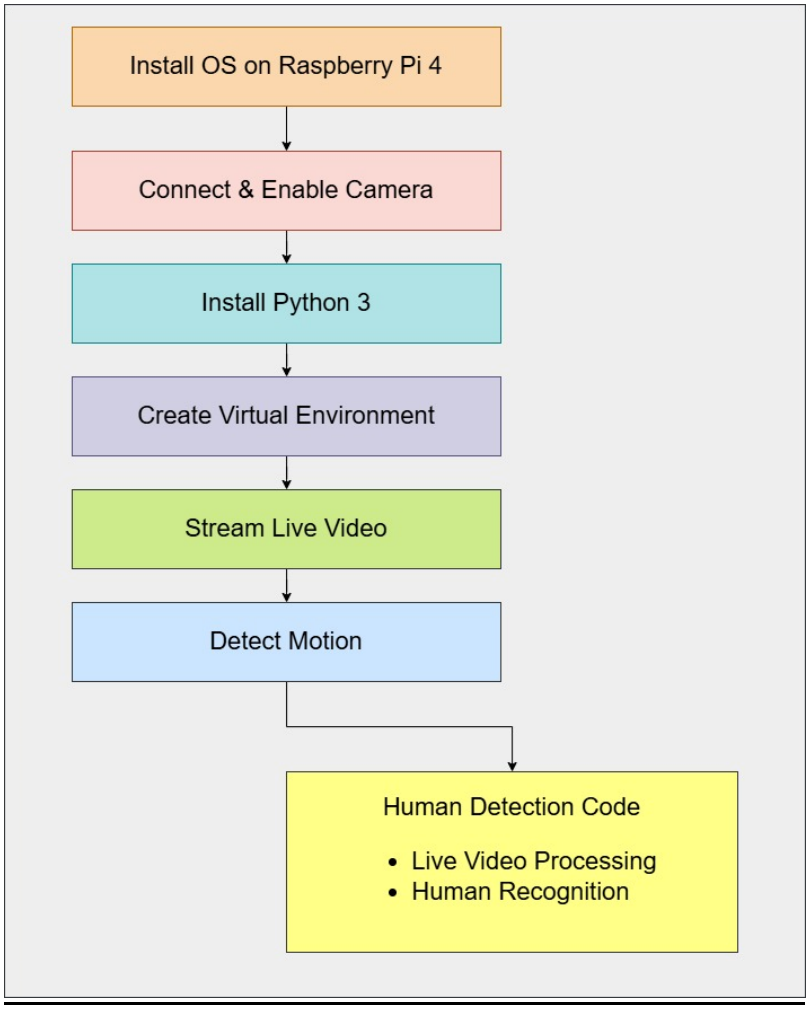

Fig 1: Progress of FallGuard

Tasks Completed

Hardware Setup:

- Integrated the Raspberry Pi with a webcam for real-time video capture.
- Configured the camera to ensure stable connectivity.

Software Implementation:

- Installed OpenCV and configured the Raspberry Pi environment.
- Implemented HOG + SVM-based human detection.
- Optimized the camera module to handle reconnection issues.

Fall Detection Dataset Preprocessing

- Found a suitable fall detection dataset from Kaggle.
- Analyzed dataset attributes, including image/video frames, labels, and resolution.
- Cleaned and preprocessed the data (resizing, normalization, grayscale conversion).
- Prepared the dataset for model training and evaluation.

Data Flow

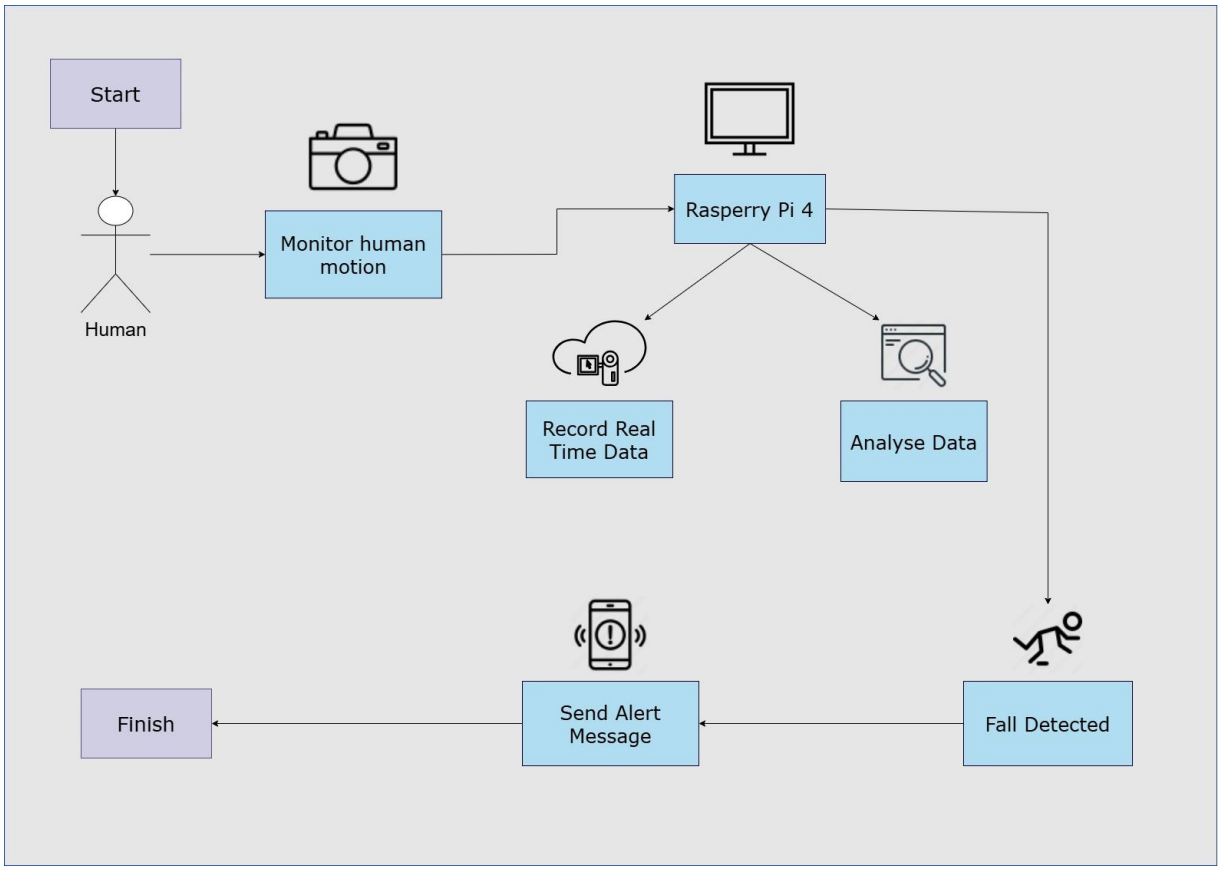

Fig 2: Work Flow Diagram of FallGuard

The system follows this workflow for real-time fall detection:

1. Monitor Human Motion
   - The webcam captures live video frames and streams them to the Raspberry Pi.
   - The system attempts multiple retries if the camera connection fails.
2. Raspberry Pi 4 Processing:
   - Frames are resized to 640x480 pixels for faster processing.
   - The frames are converted to grayscale to improve detection accuracy.
3. Human Detection:
   - The HOG + SVM model detects people in the frame.
   - The system draws bounding boxes around detected individuals.
4. Fall Detection (Planned Enhancement):
   - A deep learning model will analyze posture changes.
   - If a fall is detected, the system will trigger an alert.
5. Alert System (Future Implementation):
   - If a fall is detected, an alert is sent to caregivers via SMS.
   - Future integration with IoT for remote monitoring.

Architecture

The Fall detection system is composed of the following key components:

Hardware Components:

1. Raspberry Pi (Processing Unit)
    - Acts as the central processing unit for the device.
    - Runs the computer vision model and processes the video feed.
    - Optimized for low power consumption and continuous operation.
2. Webcam (Camera Module)
    - Captures real-time video at 640x480 resolution for analysis.
    - Connected to the Raspberry Pi via USB.
    - Used to track human movement and detect falls.
3. Power Supply
    - The system runs on a 5V power adapter, ensuring continuous operation.
    - Future improvements may include battery backup for portability.
4. External Alert System (Planned Feature)
    - It will send an alert message to caregivers on their phones.

Software Components:

1. Operating System: Raspberry Pi OS (Linux-based).
2. Programming Language: Python.
3. Libraries & Frameworks:
    - OpenCV: For video processing, real-time image analysis, and human detection.
    - HOG + SVM Detector: Used for identifying humans in the frame by extracting features and classifying them.
    - NumPy: For numerical computations, image processing, and array manipulations.
    - pandas: For handling and analyzing structured data.
    - Matplotlib: For visualizing data and plotting graphs.
    - TensorFlow Lite (TFLite): For running deep learning models efficiently on edge devices like Raspberry Pi.
    - Torchvision: Provides datasets, models, and transforms for image processing with PyTorch.
4. Future Integration: Implement deep learning-based models (e.g., TensorFlow, PyTorch) to enhance accuracy and robustness in human detection.

Challenges

1. Human Detection Accuracy:
    - Differentiating between actual falls and similar movements (e.g., sitting down quickly).
    - Reducing false positives and negatives.
2. Camera & Processing Limitations:
    - The webcam's low-light performance affects detection.

- Real-time processing on Raspberry Pi needs optimization for better performance.

3. Privacy Concerns:
   - Since the system uses a camera, data privacy must be addressed.
   - Exploring edge computing to avoid cloud-based data storage.
4. Alert System Integration:

- Need to develop a reliable method to notify caregivers immediately when a fall is detected.

Conclusion

FallGuard enhances elderly safety by detecting falls in real time using Raspberry Pi and computer vision. It alerts caregivers instantly, ensuring quick response. Future improvements will focus on accuracy, real-time processing, and IoT integration for better reliability.

Contribution:

Member ID	Primary Contribution	Secondary Contribution
20200204073	Coding	Report Writing
20200204092	Report Writing	Hardware
20200204104	Report Writing	Coding
20200204106	Hardware	Coding