FallAlert: A Mobile Application for Real-Time Fall Detection and Emergency Assistance

Project Final Report (Total Words Count: 1550)

Om Bhosale (1293903), Xunzhong Qin(1307098)
ENGG*6400-Mobile Device App Development -S24
University of Guelph

Abstract

As the problem of population aging becomes more and more serious in many countries, this project addresses a significant need for safety and security among senior citizens, leveraging smart device technology to enhance their independence and wellbeing.

Keywords: Smart Device Sensors, Emergency Call, Daily Activity, Fall Detection, Android App, Machine Learning, Neural Network.

1 INTRODUCTION

According to the World Health Organization (WHO) reports, falls are prominent among the external causes of unintentional injury. Fall-related injuries are more common among older persons and are a major cause of pain, disability, loss of independence and premature death. Approximately 28-35% of people aged of 65 and over fall each year increasing to 32-42% for those over 70 years of age.

In some cases, fall will cause people lose the capability to ask for emergency help. Without medical help this person might lost live，especially for the people who lives or works alone. This project aims to design and implement an Android application to detect the falls for elderly individuals or the person who works in dangerous environments, providing immediate assistance by making an emergency call or sending a text message to pre-configured contacts upon detecting a fall. It will significantly reduce the emergency response times and increase people's chance of survival.

2 OVERRALL DESIGN

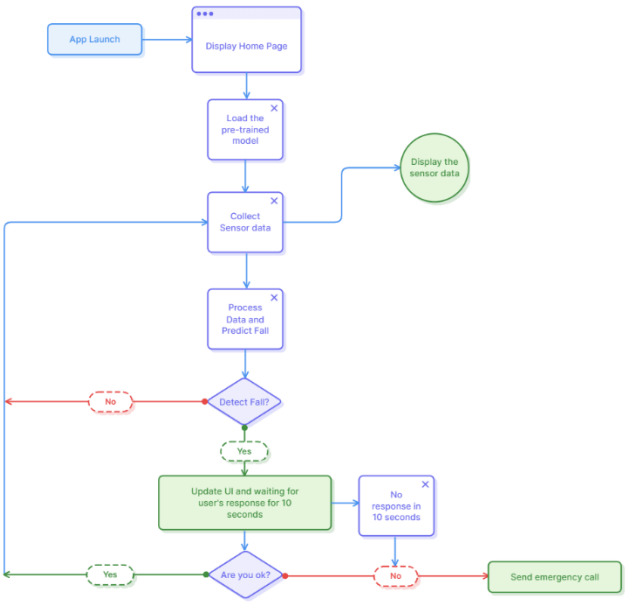
The solution for fall detection can be divided into three groups by using different devices: system rely on smart device with various sensors such as accelerometer, system rely on monitor the person by external devices such as camera, and system that hybridize these two approaches. This project chose developing an Android App to detect the fall by using integrated accelerometer and gyroscope sensor.

For the algorithm of fall detecting, it can be divided into two categories, one is threshold-based, and another is Machine-learning based. In this project we used Machine-learning based method.

The overall design includes two parts, one is the App and another is the AI model. This report will focus on the App design and development.

2.1 App overall design

The workflow of FallAlert app is showing as below block diagram. It has 3 classes, one is the main activity class, one is a separate class to handle the sensor data buffer, and one is the class to handle the model prediction.



Display Home Page: In the main activity layout, using Textview to show the accelerometer and gyroscope sensor data in realtime, including acceleration force along the X,Y,Z axis and rate of rotation around X,Y,Z axis. Also, using Textview the showing the fall alert message when a fall is detected. Three buttons are placed on the home page, "Call Caregiver", "Call Emergency" and "Setup". User click "call caregiver" to call the care giver, click "Call emergency" to contact 911

emergency center, and the "Setup" is designed to let user configure the features and their personal information.

Load the pre-trained model: This model is organized in a separated class file, using tensorflow.lite.interpreter library to load the pretrained model from file named "fall_detection_model.tflite".

Collect sensor data: This model will use android.hardware library to collect the data from accelerometer and gyroscope sensor.

Process data and predict the fall: The data are store in two Circular buffer by using CircularArray. In the next phase, the raw data from sensor will be preprocessed, filter out noise, normalized, etc. Currently the app only normalizes the sensor data to match the input of the model. The preprocess data is then fed into the model to do prediction. The model will return a predictionScore. If the score is greater than 0.5 (This value can be adjusted), app will consider it as a fall. Major functions are implemented in the onSensorChanged event.

Emergency response: Once a fall is detected, the App will show the alert message on the home page, then popup an alert dialog to ask if the user ok or not and wait for user's response, at the meantime, a time counter is activated, time click number will show on the alertDialog. The app will implement the action according to user's response, like ignore the fall or call emergency. During the timer click period, if the user has no response, FallAlert will call the emergency automatically.

2.2 Machine-Learning Model overall design

The model is a Sequential neural network model defined with three layers. An input layer with 64 neurons and ReLU activation, A hidden layer with 32 neurons and ReLU activation, and an output layer with 2 neurons and Softmax activation for binary classification.

The model is trained for 2 epochs using a batch size of 32, with validation on the test set.

After training and validation successfully, the trained model is saved in TensorFlow's SAVEDMODEL format and then converted to TensorFlow Lite format for deployment on Android devices.

3 FUNCTIONALITY

The FallAlert app successfully demonstrates key functionalities essential for ensuring the safety and well-being of elderly individuals or those working in hazardous environments. The primary functionalities implemented and tested include:

1. Real-time Fall Detection:

The app uses accelerometer and gyroscope data to continuously monitor the user's movements. The sensor data is fed into a machine learning model to detect falls accurately.

When a fall is detected, the app triggers an alert, asking the user if they are all right.

2. Emergency Call Initiation:

Upon detecting a fall and confirming that the user needs assistance, the app can automatically initiate an emergency call to a pre-configured contact.

Users can configure this emergency contact within the app settings, ensuring that help is contacted promptly during an emergency.

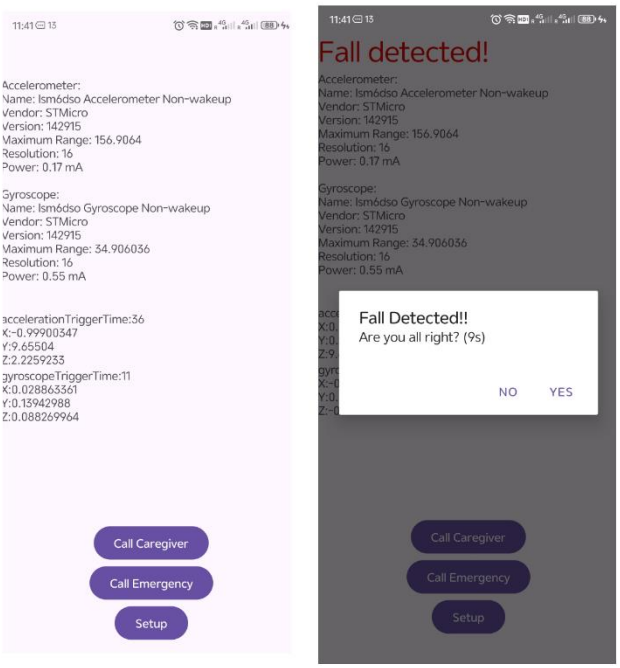
3. User Interaction:

The app displays sensor data in real-time, allowing users to see the accelerometer and gyroscope readings.

When a fall is detected, an alert dialog pops up, asking the user if they are all right. The user can respond by selecting either "Yes" or "No."

If the user confirms they are not all right, the app will initiates an emergency call.

Screenshots:



Main Activity: Displays real-time sensor data and buttons for calling a caregiver, emergency contact, and accessing setup.
Alert Dialog: A pop-up alert asking the user if they are all right upon detecting a fall.
Issues Encountered:
Some false positives were observed during initial testing, where normal activities were incorrectly identified as falls. This was mitigated by refining the model and adjusting the sensitivity.
Ensuring that the alert system does not display multiple alerts simultaneously required careful implementation of the alert tracking flag.

4 FUTURE WORK

For this App, the major concern is the accuracy and reliability of the fall detection, as well as determining whether an emergency call should be launched. In this project, we finished the major process and function including fall detection and emergency call launching. For further research, we can compare more training data from different resources, and implement more different fall detect approaches to find the most accurate method. User friendly interface interaction is another important part that we can improve, like showing the sensor data as a dynamic graph, implement the setup model to let user personalize the feature and function, preconfigure their personal information, store the history data, upload the data to the cloud for further training process etc.

5 SUMMARY

The FallAlert app addresses a critical need for enhancing the safety and independence of elderly individuals and those working in hazardous environments. By leveraging the sensors available in smart devices, the app effectively monitors for falls and provides immediate assistance when needed. The integration of machine learning ensures accurate detection, while the user interface facilitates prompt emergency response.

Key Learnings:
Sensor Data Utilization: Understanding and processing sensor data from accelerometers and gyroscopes were crucial for accurate fall detection.
Machine Learning Implementation: Implementing and optimizing a machine learning model for fall detection on an Android device provided valuable insights into on-device inference and performance optimization.
User Interaction Design: Ensuring that the app is user-friendly and responsive was essential, especially in emergency scenarios where quick interaction is required.

Future Enhancements:
Improving Accuracy: Collecting more diverse training data and experimenting with different machine learning models could further improve fall detection accuracy.
Enhanced User Interface: Adding features such as dynamic graphs for sensor data, user personalization options, and a more intuitive setup process.
Cloud Integration: Storing sensor data and user history in the cloud for further analysis and model training, providing more robust and personalized fall detection capabilities.
By addressing these future enhancements, the FallAlert app can become a more comprehensive and reliable tool for ensuring the safety and well-being of its users.

6 REFERENCE

1. Dariusz Mrozek, Anna Koczur, Bo_zena Małysiak-Mrozek. “Fall detection in older adults with mobile IoT devices and machine learning in the cloud and on the edge”. (2020)
2. Thiago B. Rodrigues, Débora P. Salgado, Mauricio C. Cordeiro, Katja M.Osterwald, Teodiano F. B. Filho “Fall Detection System by Machine Learning Framework for Public Health” (2018)
3. Arkham Zahri Rakhman, Widyawan Widyawan, Lukito Nugroho, Kurnianingsih Kurnianingsih. “Fall Detection System Using Accelerometer and Gyroscope Based on Smartphone” (2014)
4. K. Yildirim, G. Ucar, T. Keskin, and A. Kavak, “Fall Detection Using Smartphone-Based Application(2016)
5. Islam, M.M., Neom, N.H., Imtiaz, M.S., Nooruddin, S., Islam, M.R., Islam, M.R. “A review on fall detection systems using data from smartphone sensors” (2019).