

LEBANESE  
INTERNATIONAL  
UNIVERSITY



# Fall Detection System using Computer Vision

Senior Project

By

**Ali Osseili and Malak Alkazem**

Submitted to the School of Arts & Science of the  
Lebanese International University

In part of fulfillment of the requirements for  
the degree of

**BACHERLOR OF SCIENCE IN  
COMPUTER SCIENCE AND INFORMATION  
TECHNOLOGY**

**Supervised by: Dr. Youssef Roumieh**

Spring 2024

## Acknowledgment:

We would like to express our sincere gratitude to the following individuals and entities whose contributions were instrumental in the completion of this senior project:

Dr Youssef Roumieh, PhD., our project advisor, for his invaluable guidance, insightful feedback, and unwavering support throughout the research and development process. His expertise in software development and deployment were crucial in shaping the direction of this project and ensuring its successful outcome

Sarah El-Rashidi, Ms., for providing valuable technical assistance and industry insights during the development phase. Her practical advice and support helped us overcome technical challenges and implement the project effectively.

Of course, I can't forget my family. Their unwavering support, like a well-meaning but ultimately distracting cheerleader, ensured this project got the full attention it deserved. While their constant inquiries about "how's it going?" and "is it done yet?" were certainly...motivational, let's just say uninterrupted focus might have shaved off a good eight weeks. But hey, family is family, and who needs deadlines anyway, right? We finished eventually, and that's what counts!

## Abstract:

This project presents a novel approach to fall detection using computer vision for elderly care or assisted living environments. The system leverages a YOLO (You Only Look Once) object detection model, trained on a custom dataset to identify human figures within a video stream. This lightweight model facilitates deployment on resource-constrained devices, making it accessible for a wider range of users.

The real-time video analysis is implemented as an API, enabling integration with a mobile application. Firebase serves as the communication platform, triggering notifications on designated mobile devices whenever a fall event is detected.

This project contributes to improved elder care by providing a non-intrusive, vision-based system for fall detection. The focus on efficient model design and cloud-based communication ensures broader applicability and scalability for real-world scenarios.

## Table of Figures

Figure 1: A screenshot showing the home screen of the mobile application .....	16
Figure 2: A screenshot showign the Connect Your Application screen .....	17
Figure 3: A screenshot showing the Read Me screen.....	18
Figure 4: A screenshot showing the settings screen .....	19
Figure 5: A screenshot showing the Location screen .....	20
Figure 6: A screenshot showing the Test a Fall screen .....	21
Figure 7: A screenshot showing the Fall Detected Screen .....	22
Figure 8: A screenshot showcasing a part of the dataset that was used to train the model .	26
Figure 9: A screenshot showing the image annotation data .....	27
Figure 10: An image showing different metrics as our model trained .....	28

## Table of Contents

<b>ACKNOWLEDGMENT:</b>	<b>2</b>
<b>ABSTRACT:</b>	<b>3</b>
<b>TABLE OF FIGURES</b>	<b>4</b>
<b>CHAPTER 1</b>	<b>6</b>
1.1 PROBLEM STATEMENT	6
1.2 PROPOSED SOLUTION	6
1.3 TECHNOLOGY CONSTRAINTS	6
1.4 PROJECT OBJECTIVES	6
<b>CHAPTER 2: REQUIREMENTS ANALYSIS</b>	<b>7</b>
2.1 FUNCTIONAL REQUIREMENTS	7
2.2 USE CASE DIAGRAM	8
2.2 NON-FUNCTIONAL REQUIREMENTS	9
2.3 UML USE CASES	9
<b>CHAPTER 3: MOBILE APPLICATION AND DESIGN</b>	<b>15</b>
3.1 HOME SCREEN	16
3.2 CONNECT YOUR APPLICATION	17
3.3 READ ME	18
3.4 SETTING SCREEN	19
3.5 GET YOUR LOCATION SCREEN	20
3.6: TEST A FALL	21
3.7 FALL DETECTED:	22
3.8 MOBILE APP USE CASES	23
3.9 FUTURE MOBILE APP UPDATES	24
3.10 CONCLUSION	25
<b>CHAPTER 4: THE MODEL</b>	<b>25</b>
4.1 INTRODUCTION	25
4.2 DATA COLLECTION	25
4.3 DATA ANNOTATION	26
4.4 PREPROCESSING AND AUGMENTATION	27
4.5 ACCURACY AND METRICS	27
<b>CHAPTER 5: CONNECTING INTELLIGENCE TO ACTION - MOBILE APP AND FALL DETECTION MODEL</b>	<b>28</b>
5.1. SYSTEM OVERVIEW	28
5.2. COMMUNICATION BRIDGE: FIREBASE	28
5.3. MOBILE APP IN ACTION	29
5.4. BENEFITS OF REAL-TIME CONNECTION	29
5.5. CONCLUSION	29
<b>CHAPTER 6: CONCLUSION - A SAFETY NET FOR THE VULNERABLE</b>	<b>29</b>
6.1 IMPACT AND FUTURE CONSIDERATIONS	30
6.2 A FINAL NOTE	30

## Chapter 1

### 1.1 Problem Statement

Falls are a significant health concern, particularly for the elderly population. According to the Centers for Disease Control and Prevention (CDC), falls are the leading cause of injury deaths among adults aged 65 and older, resulting in over 32,000 fatalities annually [1]. Furthermore, falls can lead to serious injuries like hip fractures, which can significantly impact an individual's mobility and independence.

Traditional fall detection methods often rely on wearable sensors, which can be intrusive, uncomfortable, or forgotten by users. This project addresses the need for a non-intrusive and user-friendly fall detection system suitable for home environments.

### 1.2 Proposed Solution

This project proposes a real-time fall detection system utilizing computer vision and mobile notifications. The system leverages a lightweight YOLO (You Only Look Once) object detection model trained on a custom dataset of human figures. This model can be deployed as an API, enabling integration with a mobile application. Upon detecting a fall event in the video stream, the system triggers a notification sent through Firebase to android mobile devices with the application, alerting caregivers or family members.

### 1.3 Technology Constraints

While computer vision offers a promising approach to fall detection, certain constraints need to be considered:

**Accuracy:** Accurately distinguishing falls from other activities like sitting or kneeling is crucial.

**Computational Cost:** Running complex models on resource-constrained mobile devices can be challenging.

**Privacy Concerns:** Video-based systems raise privacy considerations, requiring careful design and data handling practices.

### 1.4 Project Objectives

This project aims to develop a fall detection system that addresses the aforementioned constraints. The specific objectives include:

Develop a YOLO model with high accuracy in identifying falls within a video stream.

Optimize the model for efficient operation on resource-constrained devices.

Using Roboflow's API for real-time video analysis, enabling integration with a mobile application.

Utilize Firebase for secure communication and notification delivery when falls are detected.

By achieving these objectives, this project aims to provide a user-friendly and effective fall detection solution for elderly care and assisted living environments.

Reference:

[1] Centers for Disease Control and Prevention (CDC). (2023, March 15). Falls Among Older Adults. Retrieved from <https://www.cdc.gov/falls/index.html>

## Chapter 2: Requirements Analysis

This chapter outlines the functional and non-functional requirements for the fall detection system. Additionally, it details the UML use cases and scenarios to illustrate the system's functionalities from a user perspective.

### 2.1 Functional Requirements

Functional requirements define the specific actions the system should perform. Here are some key functionalities:

- **Fall Detection:** The core functionality of the system is to analyze the video stream in real-time and detect fall events with high accuracy.
- **Emergency Contact Selection:** Users can designate emergency contacts to be notified in case of a fall.
- **Local Emergency Number:** The system automatically dials the local emergency number based on the user's location to connect with emergency services.
- **Fall Test:** Users can initiate a simulated fall scenario to test the system's detection capabilities and ensure proper operation.
- **Response Time Selection:** Users can define a delay period before an emergency call is initiated, allowing for potential intervention by a caregiver present at the scene.
- **Ringtone Selection:** Users can personalize their notification ringtone for fall detection alerts.
- **Video Streaming:** The system captures video data from the camera and transmits it for processing.
- **Notification Delivery:** The system sends notifications to designated recipients (mobile app) upon detecting a fall event.
- **Emergency Call Cancellation:** Residents have the option to cancel an emergency call within the designated response time window.

## 2.2 Use Case Diagram

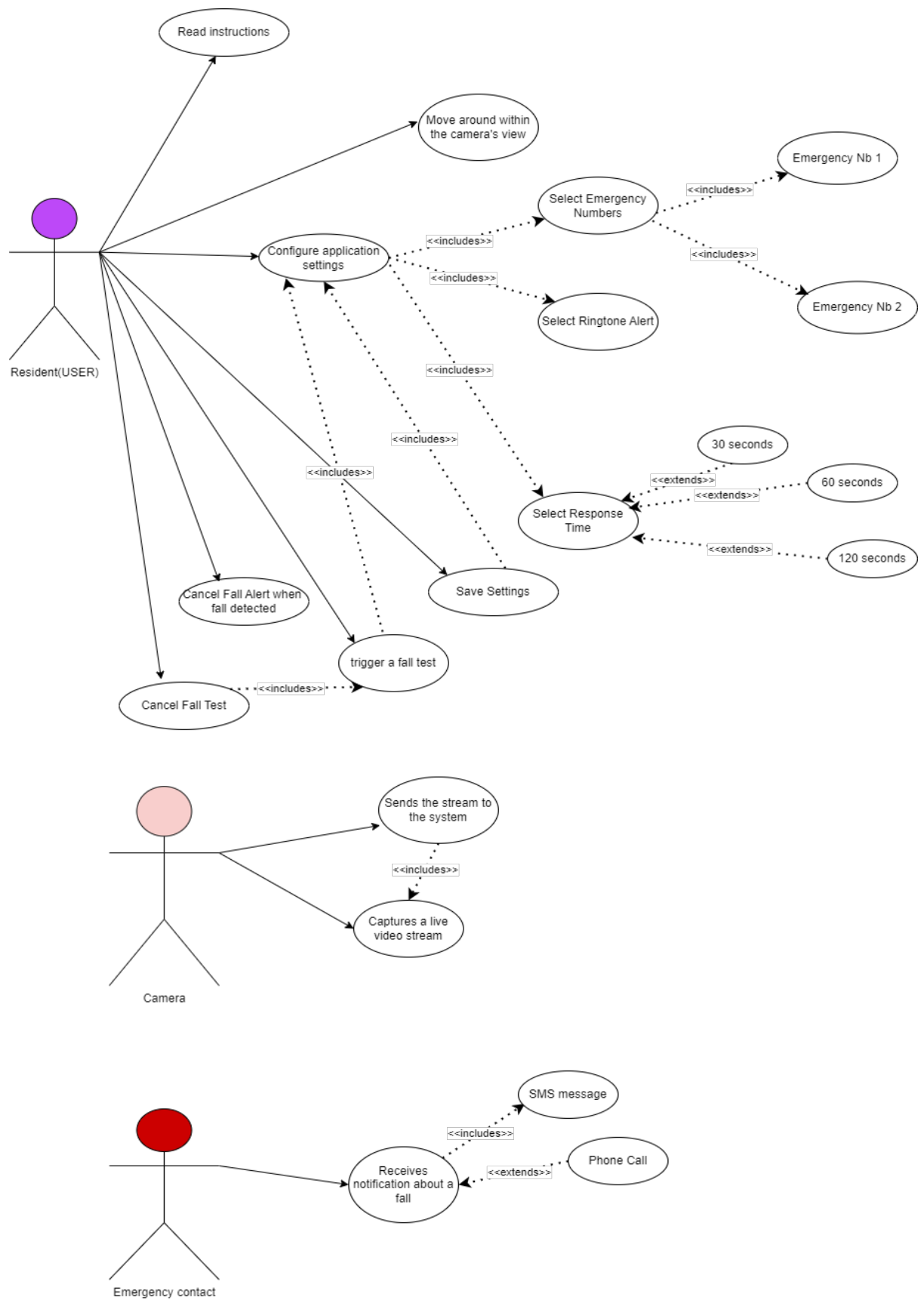


Figure 1: Usecase Diagram



## 2.2 Non-Functional Requirements

Non-functional requirements define overall system characteristics:

- **Fall Detection Accuracy:** The system should have a high true positive rate for fall detection, minimizing false positives (alerts for non-fall events).
- **Data Security:** The system must securely store and transmit user data, including video streams and emergency contact information. Adherence to data privacy regulations is crucial.
- **Response Time:** There should be minimal delay between fall detection and sending emergency notifications or initiating an emergency call.
- **Scalability:** The system architecture should allow for adding more users and cameras without impacting performance.
- **Ease of Use:** The system should be user-friendly for installation, configuration, and operation, catering to users with varying technical skill levels.
- **Privacy:** The system should have clear and customizable privacy settings. Users should have control over data collection, storage, and usage.

## 2.3 UML Use Cases

UML (Unified Modeling Language) use cases represent system functionalities from the user's perspective. Here are some key use cases for our fall detection system:

<b>Use-Case Name:</b>	Configure application settings
<b>Use-Case ID:</b>	AT-001
<b>Priority:</b>	High
<b>Primary Actor:</b>	Resident
<b>Description:</b>	This use case describes the scenario where a resident configures the mobile application settings for fall detection. This includes emergency contact information, notification preferences, and response time.
<b>Precondition:</b>	The resident has downloaded and installed the fall detection mobile application on their smartphone. The resident has launched the application and accepted the permissions

Typical Course of Events:	Actor Action	System Response
	<ol style="list-style-type: none"> <li data-bbox="550 253 960 510">1. The resident launches the application for the first time. <b>If storage permissions are already granted</b> The resident taps on the "Settings" menu option within the application.</li> <li data-bbox="550 981 960 1272">2. The resident chooses one of the following: <b>Allow:</b> The resident grants storage permission to the application. <b>Deny:</b> The resident denies storage permission to the application.</li> <li data-bbox="550 1317 960 1417">3. The resident taps on the "Settings" menu option within the application.</li> <li data-bbox="550 1529 960 1630">4. The resident taps on the "Emergency Contacts" section.</li> <li data-bbox="550 1709 960 1843">5. The resident Add a new emergency contact by selecting him from contacts list opened.</li> <li data-bbox="550 1921 960 2022">6. The resident taps on the "Notification Preferences" section.</li> </ol>	<ol style="list-style-type: none"> <li data-bbox="973 253 1469 902">1. The application checks for necessary storage permissions (contacts and ringtones). <b>If storage permissions are already granted</b> The application displays the settings screen with various configuration options (Step #2 from original use case). <b>If storage permissions are not granted:</b> <ul style="list-style-type: none"> <li data-bbox="1029 656 1461 835">• The application displays a notification requesting permission to access storage for storing emergency contacts and ringtones.</li> <li data-bbox="1029 835 1461 902">• The notification provides options for "Allow" and "Deny".</li> </ul> </li> <li data-bbox="973 1014 1469 1261">2. If Allowed: The application hides the permission notification and proceeds to display the settings screen If Denied: The application deny the resident to enter settings page and configure settings</li> <li data-bbox="973 1328 1469 1440">3. The application displays the settings screen with various configuration options.</li> <li data-bbox="973 1552 1469 1653">4. The application displays a list of existing contacts saved on phone or a blank form to add new contacts.</li> <li data-bbox="973 1720 1469 1798">5. The application saves the contacts accordingly.</li> <li data-bbox="973 1933 1469 2045">6. The application displays options for customizing notifications upon fall detection, such as</li> </ol>

	<p>enabling/disabling ringtones, vibration alerts, and on-screen messages.</p> <p>7. The resident selects their preferred notification settings.</p> <p>8. The resident taps on the "Fall Response Time" section.</p> <p>9. The resident chooses their desired fall response time from the available options (e.g., 10 seconds, 30 seconds, etc.).</p> <p>10. The resident taps on the "Save" button to confirm all configuration changes.</p>	<p>7. The application reflects the selected notification preferences.</p> <p>8. The application displays options to set a delay before initiating a call to emergency contacts after a fall is detected. This allows the resident time to cancel the alert if it's a false positive.</p> <p>9. The application saves the selected fall response time.</p> <p>10. The application confirms the settings have been saved successfully.</p>
<b>Postconditions:</b>	<ul style="list-style-type: none"> <li>• If the resident grants storage permission, their emergency contact information, notification preferences, and fall response time are configured according to their choices within the application.</li> <li>• If the resident denies storage permission, the application might have limited functionality or return to the main screen.</li> </ul>	

2)

<b>Use-Case Name:</b>	Resident Triggers Fall Test
<b>Use-Case ID:</b>	AT-012

<b>Priority:</b>	Medium	
<b>Primary Actor:</b>	Resident	
<b>Description:</b>	This use case describes the scenario where a resident intentionally triggers a fall test using the mobile application. This allows them to verify the fall detection functionality and emergency response process.	
<b>Precondition:</b>	<ul style="list-style-type: none"> <li>• The resident has downloaded and installed the fall detection mobile application on their smartphone.</li> <li>• The resident has launched the application and configured their settings (including emergency contacts).</li> </ul>	
<b>Typical Course of Events:</b>	<b>Actor Action</b>	<b>System Response</b>
	<ol style="list-style-type: none"> <li>1. The resident taps on the dedicated "Fall Test" button within the application.</li> <li>2. The resident chooses one of the following:  <b>Yes:</b> The resident confirms they want to proceed with the fall test  <b>No:</b> The resident decides not to perform the fall test at this time.</li> <li>3. The resident, during the countdown timer, can tap on a "Cancel Test" button displayed within the application.</li> <li>4. If the countdown timer reaches zero and the resident hasn't canceled the test</li> </ol>	<ol style="list-style-type: none"> <li>1. The application displays a confirmation dialog box with a clear message like "Are you sure you want to initiate a fall test?"</li> <li>2. <b>If pressed NO:</b>  The application closes the confirmation dialog box, and the resident remains on the main screen. No test or notifications are initiated.  <b>If Yes:</b>  The application displays a countdown timer indicating the fall response time (as configured in the settings).  An SMS notification is sent to the emergency contacts informing them of a potential fall (clearly stating it's a test).  The selected alert activated on resident's mobile</li> <li>3. The application cancels the simulated fall alert.  The countdown timer stops.  No emergency call is initiated</li> <li>4. The application automatically initiates a phone call to the primary emergency contact.</li> </ol>

<b>Postconditions:</b>	<p>If the resident confirms the fall test, one SMS notification is sent to emergency contacts, and potentially one emergency call is attempted (depending on if the resident cancels before the timer ends).</p> <p>If the resident cancels the test, no notifications or calls are initiated.</p>
------------------------	--

3)

<b>Use-Case Name:</b>	Resident Reads Instructions	
<b>Use-Case ID:</b>	AT-114	
<b>Priority:</b>	Low	
<b>Primary Actor:</b>	Resident	
<b>Description:</b>	<p>This use case describes the scenario where a resident accesses and reviews the instructions or tutorials within the fall detection mobile application. This can include information about the application's functionalities, proper usage, and potential limitations.</p>	
<b>Precondition:</b>	<ul style="list-style-type: none"> <li>The resident has downloaded and installed the fall detection mobile application on their smartphone.</li> <li>The resident has launched the application.</li> </ul>	
<b>Typical Course of Events:</b>	<b>Actor Action</b>	<b>System Response</b>
	<ol style="list-style-type: none"> <li>The resident taps on a dedicated "Instructions" or "Help" button within the application.</li> <li>The resident reads through the instructions, focusing on topics that interest them, such as: How to configure emergency contact information and notification preferences. Understanding the fall detection process and response time.</li> </ol>	<p>The application displays a screen or opens a document containing detailed instructions on various aspects of the fall detection system. This information could be presented in different formats like text, images, or even short instructional videos.</p>

	<p>Troubleshooting common issues or limitations of the application.</p> <p>3. The resident might choose to navigate through different sections of the instructions using provided menus or buttons.</p> <p>4. The resident taps on a "Close" or "Back" button when they have finished reviewing the instructions.</p>	<p>The application closes the instructions screen and returns the resident to the previous screen they were on.</p>
<b>Postconditions:</b>	<p>The resident gains a better understanding of the fall detection application's functionalities and proper usage through the provided instructions.</p>	

4)

<b>Use-Case Name:</b>	Emergency Contact Receives Fall Alert
<b>Use-Case ID:</b>	AT-005
<b>Priority:</b>	High
<b>Primary Actor:</b>	Emergency Contact
<b>Description:</b>	This use case describes the scenario where a designated emergency contact receives a notification and potentially a phone call due to a fall detection event triggered by the resident.
<b>Precondition:</b>	<ul style="list-style-type: none"> <li>The resident has downloaded and installed the fall detection mobile application on their smartphone.</li> <li>The resident has configured the application settings, including adding the emergency contact's phone number.</li> <li>A fall event is detected by the Model</li> </ul>

<b>Typical Course of Events:</b>	<p>. The fall detection model notifies the application on the resident's phone and registers a possible fall.</p> <p>The emergency contact receives an SMS notification from the fall detection application.</p> <p>Upon receiving the notification or phone call, the emergency contact takes necessary actions based on the situation, such as:          Calling the resident's phone number to check on their well-being.          Dispatching help to the resident's location (if the location information is provided).          Contacting other family members or emergency services depending on the urgency of the situation.</p>	<p>1)The application initiates its emergency response protocol, which might include:          Sending an SMS notification to the emergency contacts listed in the app's settings.          Initiating an automated phone call to the primary emergency contact (depending on application configuration).</p> <p>2)The notification clearly informs the contact about a potential fall incident involving the resident. It might include details like:          A message stating, "Fall Alert: [Resident's Name] may have fallen."          The resident's location information (if enabled and permitted by the resident's phone settings).          A link to a map app showing the resident's location (optional).</p>
<b>Postconditions:</b>	The emergency contact is alerted about a potential fall incident involving the resident, allowing them to take appropriate action to ensure the resident's safety.	

## Chapter 3: Mobile Application and Design

This chapter shows the mobile application design and features.

### 3.1 Home Screen

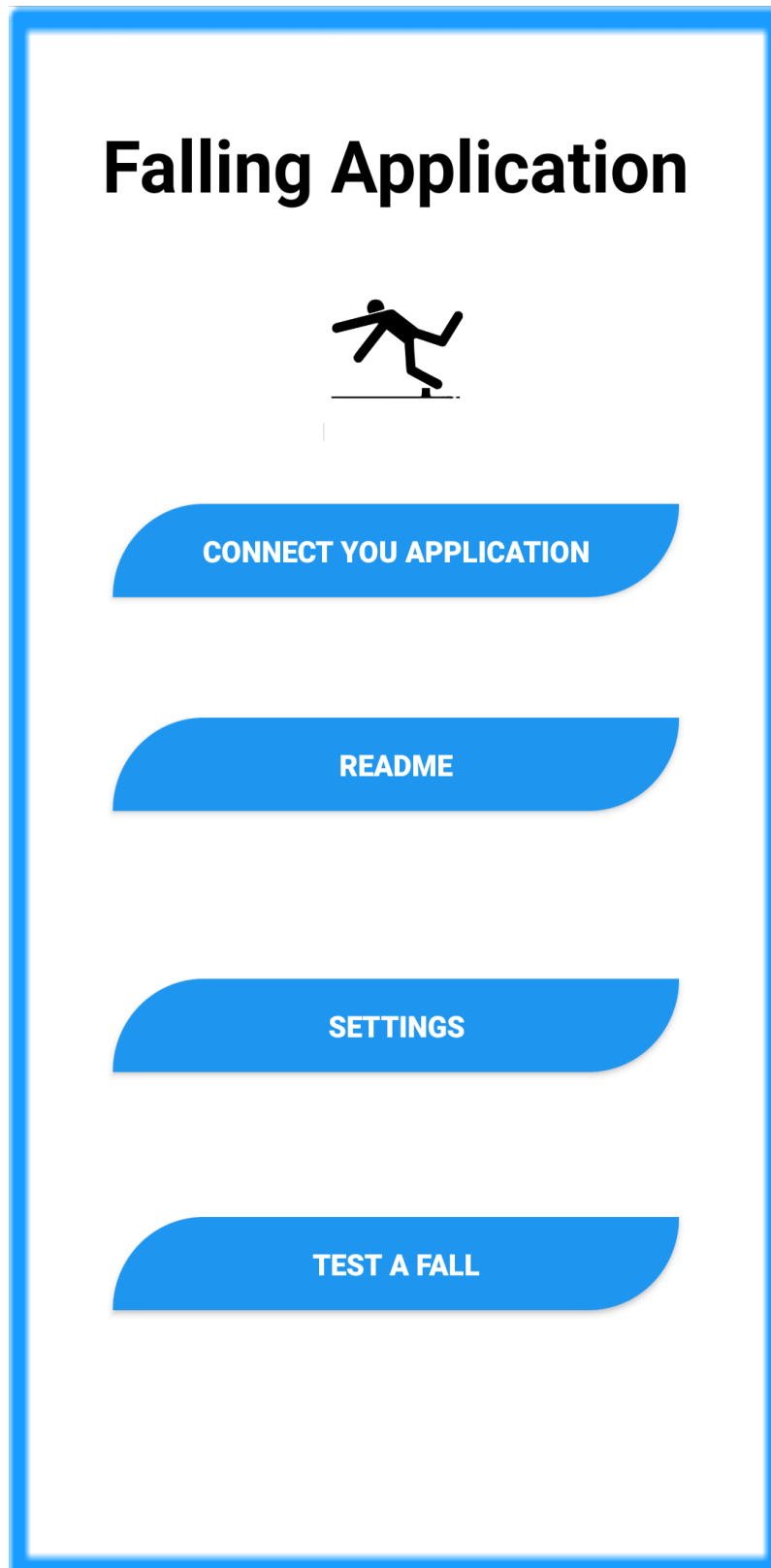
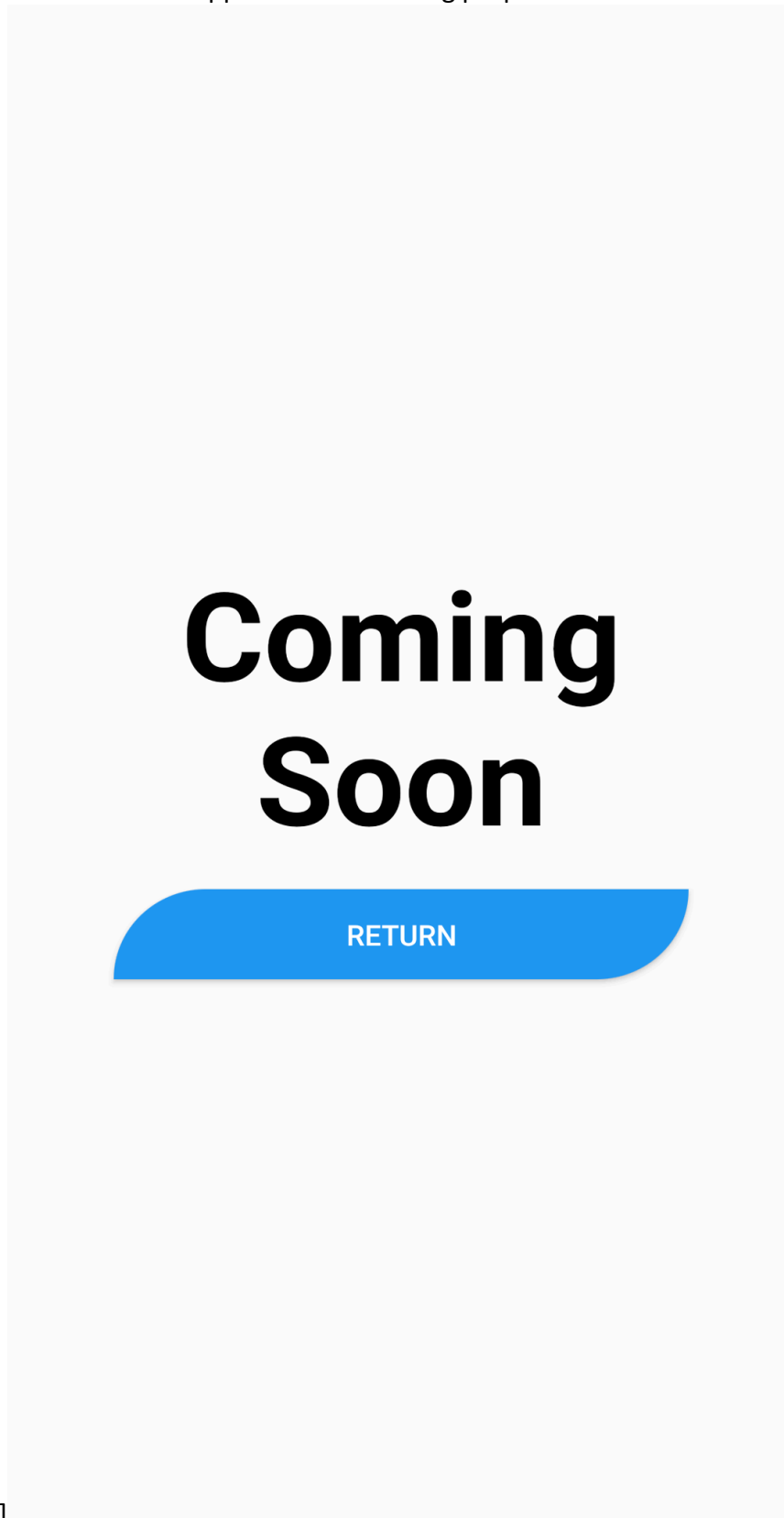


Figure 2: A screenshot showing the home screen of the mobile application



### 3.2 Connect Your Application

This feature is to be added in a future update, as for now the model will connect to all instances of the application for testing purposes.



]

Figure 3: A screenshot showign the Connect Your Application screen

### 3.3 Read Me

This screen shows the instructions for using the app, and will contain any new announcements or updates.

## Read Me

Welcome to Fall App Detection! This app is designed to help detect falls using the camera of your home and provide timely assistance in case of emergencies.

#### Steps to Use:

1. Install the app from the Google Play Store.
2. Set up your emergency contacts and customize settings.
3. Carry your device with the app running in the background.

#### How it Works:

Fall App Detection uses advanced algorithms to analyze video frames captured by your device's camera in real-time. By detecting patterns associated with falls, the app can promptly identify potential incidents and trigger appropriate responses.



**RETURN BACK**

Figure 4: A screenshot showing the Read Me screen

### 3.4 Setting Screen

This screen allows the user to change system settings as shown in figure 4 below.

The screenshot displays a settings interface with a light gray background. At the top, the word "Settings" is written in a large, bold, black font. Below it, there are four blue buttons with white text, each with rounded corners and a slight shadow. The first button is labeled "SELECT LOCATION". Below this button, the text "Malak" is displayed in a large, black font. The second button is labeled "SELECT EMERGENCY NB 1". Below it, the text "Select Emergency Nb 2" is displayed in a large, black font. The third button is labeled "SELECT EMERGENCY NB 2". Below it, the text "Select Ringtone Alert" is displayed in a large, black font. The fourth button is labeled "SELECT RINGTONE". Below this button, the text "Choose your Response Time:" is displayed in a large, black font. Underneath this text, the text "5 seconds" is displayed in a smaller, black font, followed by a small black downward-pointing triangle. At the bottom of the screen, there is a large blue button with white text labeled "SAVE".

Figure 5: A screenshot showing the settings screen

### 3.5 Get Your Location Screen

This screen prompts the user to give permissions to access the device location.

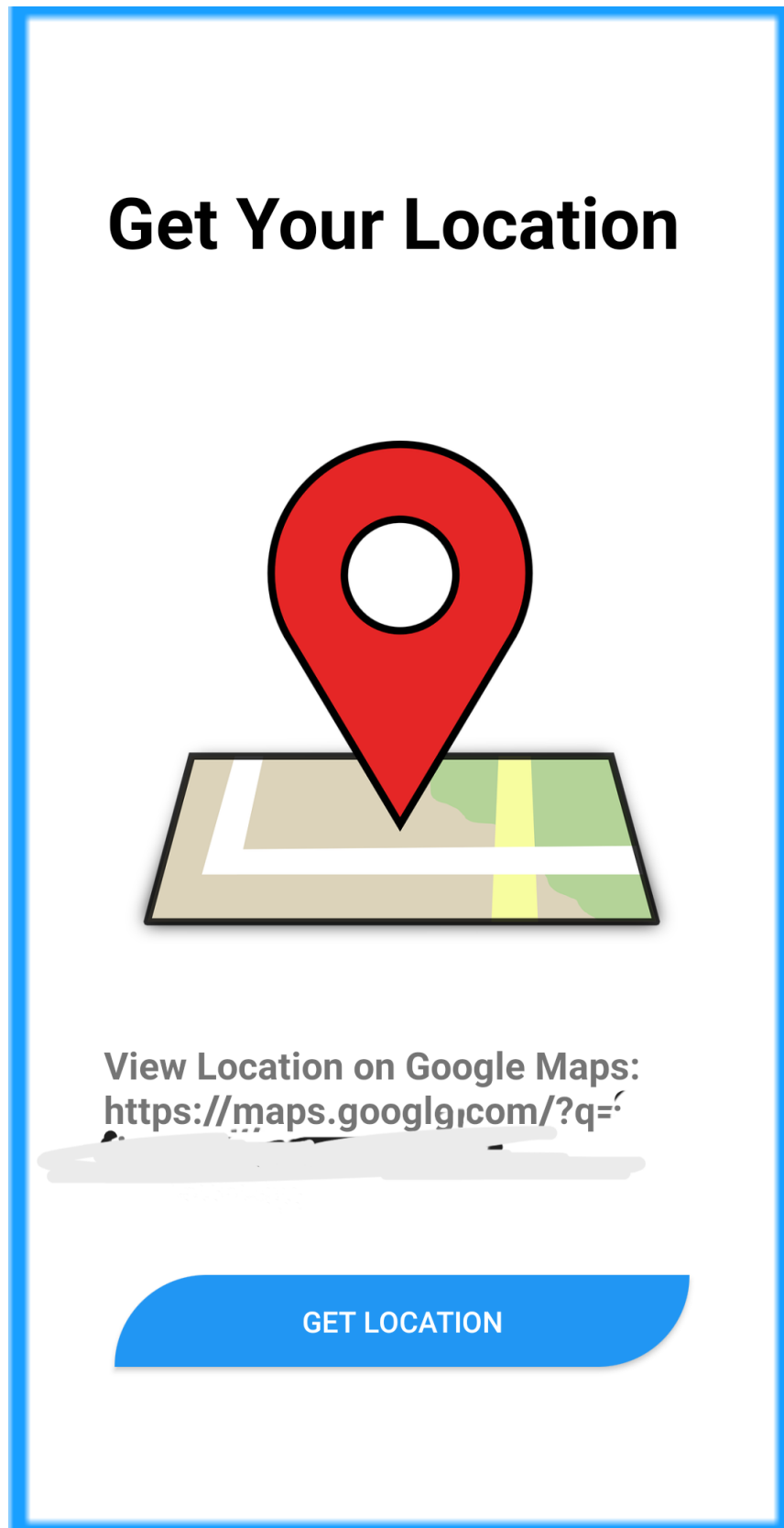


Figure 6: A screenshot showing the Location screen

### 3.6: Test a Fall

This screen allows the user to test a fall, which will trigger the countdown.

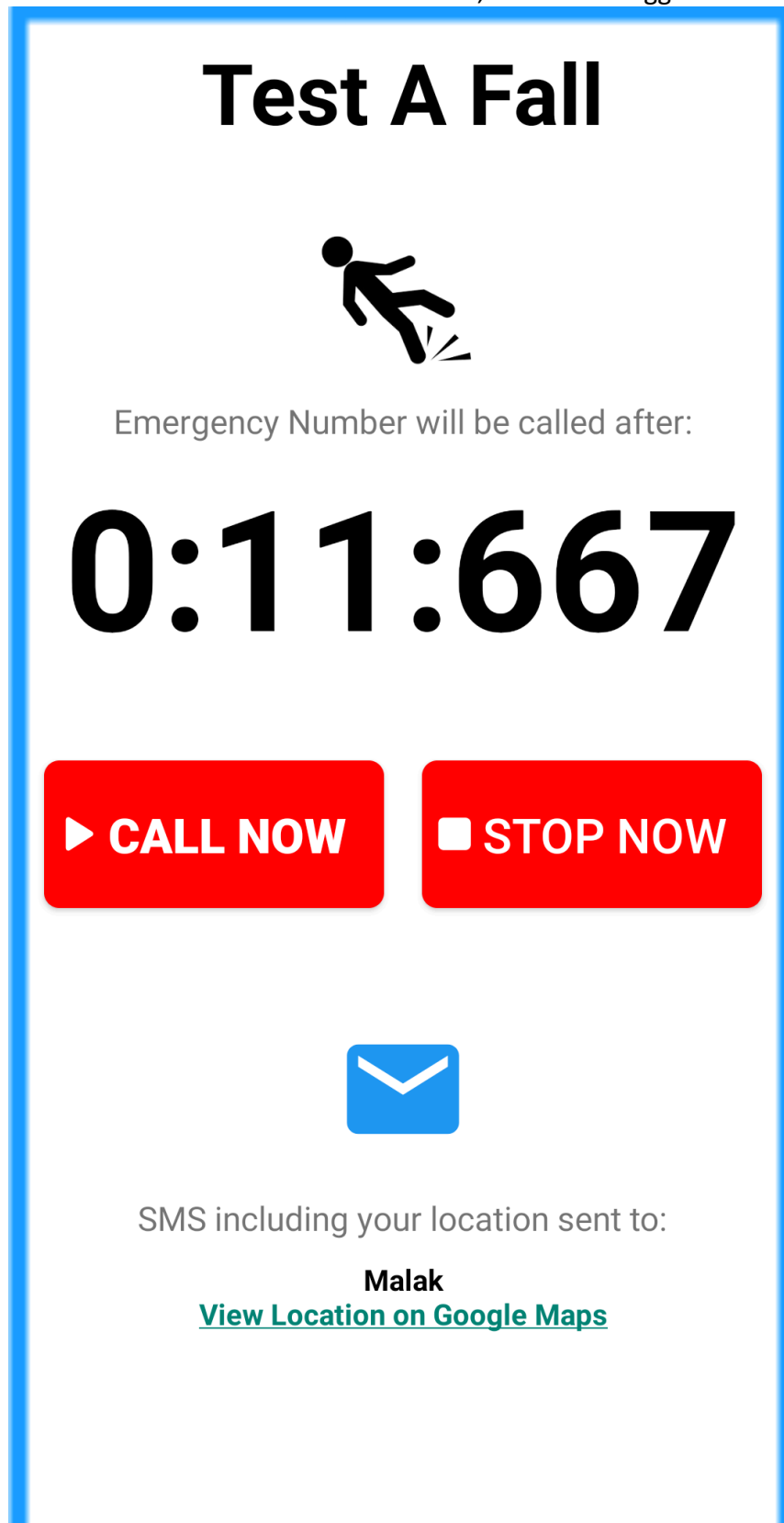


Figure 7: A screenshot showing the Test a Fall screen

## 3.7 Fall Detected:

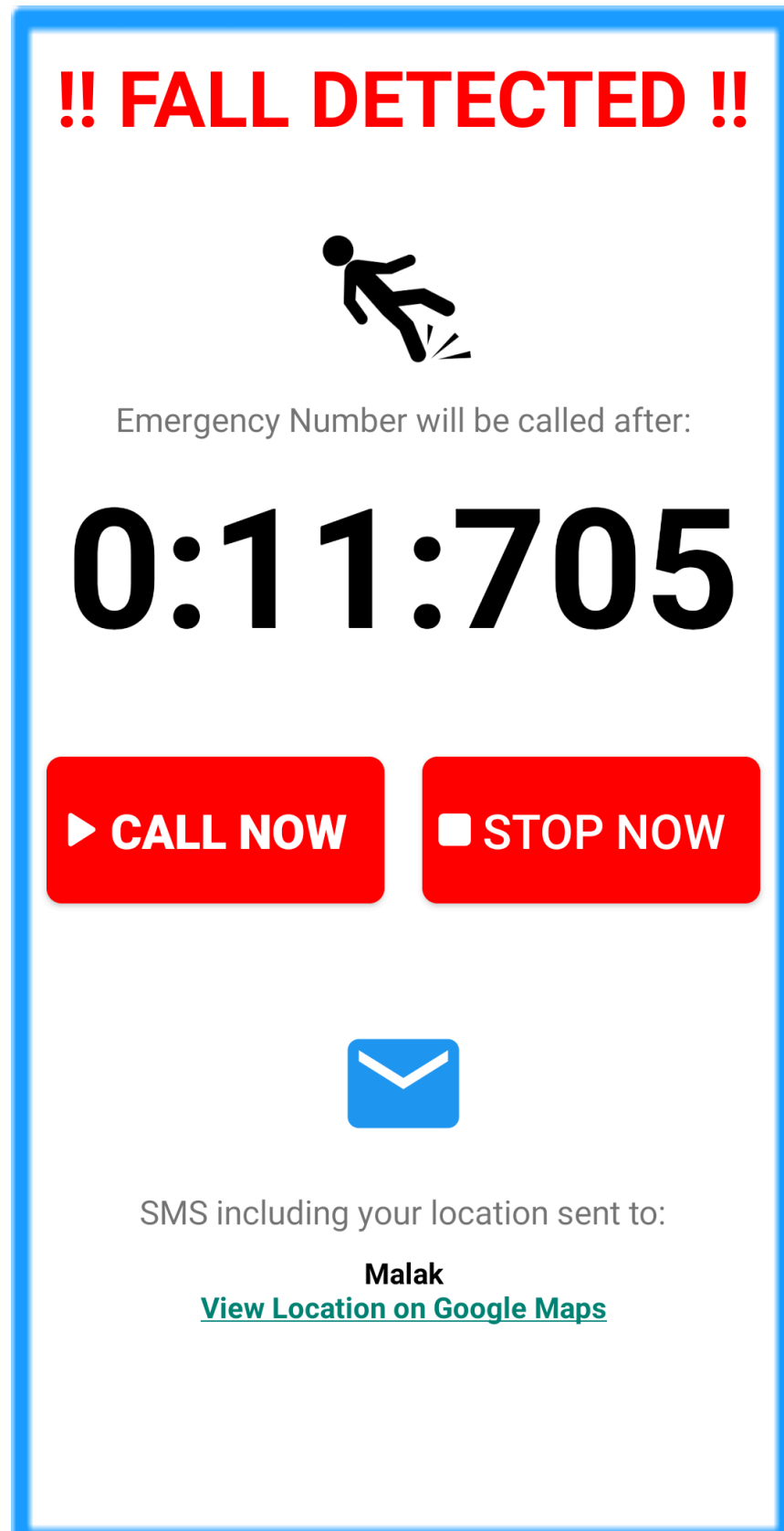


Figure 8: A screenshot showing the Fall Detected Screen

### 3.8 Mobile App Use Cases

#### 1. Select Location

Functionality:

This setting allows the user to select a specific location to be included in the SMS message sent to the emergency contact when a fall is tested or detected.

Purpose: To provide the emergency contact with precise location details of the resident, ensuring timely and accurate assistance.

Implementation:

Users choose their current location .

The selected location is stored and used in the SMS sent during emergency alerts.

#### 2. Select Emergency Number 1

Functionality:

This setting allows the user to specify the primary emergency contact number.

Purpose: To ensure that the first point of contact is someone who can respond quickly in case of an emergency.

Implementation:

The user selects a phone number from contacts list, which is saved as the primary emergency contact.

During a fall detection event, an SMS is sent to this number with the resident's location and a message indicating the emergency.

#### 3. Select Emergency Number 2

Functionality:

This setting allows the user to specify a secondary emergency contact number.

Purpose: To provide a backup contact in case the primary emergency contact is unavailable.

Implementation:

The user selects a second phone number, which is saved as the secondary emergency contact.

If the primary contact does not respond within a specified timeframe, an SMS is sent to this secondary number.

#### 4. Select Ringtone

Functionality:

This setting allows the user to select a ringtone that will play loudly when a fall is detected.

Purpose: To alert the resident or those nearby that a fall has occurred, drawing immediate attention to the emergency.

Implementation:

The user can choose from a list of pre-defined ringtones.

When a fall is detected, the selected ringtone plays at high volume on the resident's phone.

#### 5. Select Response Time

Functionality:

This setting allows the user to define the response time, which is the duration the system waits for a response before escalating the alert.

Purpose: To ensure timely escalation of the emergency alert if there is no response within the defined period.

Implementation:

The user selects a time interval (e.g., 30 seconds, 1 minute).

If no response is received within this time, the system automatically calls the emergency numbers in the sequence provided.

### 3.9 Future Mobile App Updates

#### Connect with Camera

##### Overview

The upcoming "Connect with Camera" feature will enhance the application by allowing users to quickly and easily connect to devices using a QR scanner.

This feature is designed to streamline the connection process, providing multiple options for connectivity including direct connection via QR code, Bluetooth, or a specific code.

##### Functionality

##### QR Scanner

Description: The QR scanner will enable users to scan a QR code included with their camera or other device.

Purpose: To facilitate a quick and hassle-free connection process by automatically detecting and connecting to the device.

Implementation:

Users open the QR scanner from the "Connect with Camera" page.

The scanner reads the QR code and automatically connects to the device.

##### Bluetooth Connection

Description: This option allows users to connect to devices via Bluetooth.

Purpose: To provide an alternative method of connectivity for devices that support Bluetooth pairing.

Implementation:

Users select the Bluetooth option from the "Connect with Camera" page.

The application scans for nearby Bluetooth devices and lists them.

Users select their device from the list to establish a connection.

##### Specific Code Connection

Description: Users can manually enter a specific code to connect to their device.

Purpose: To offer a manual connection method for devices that support this option, ensuring flexibility in connectivity.

Implementation:



Users choose the specific code option from the "Connect with Camera" page. They enter the code provided with their device. The application uses this code to establish a secure connection.

#### Benefits

**Ease of Use:** Simplifies the process of connecting to devices by offering multiple connection methods.

**Flexibility:** Provides users with options to choose the most convenient and compatible method for their devices.

**Speed:** Quickens the setup process, especially with the QR scanner, allowing users to connect in seconds.

This feature is currently in development and will be available in an upcoming update. Stay tuned for this enhancement, which will significantly improve the user experience by making device connections faster and more convenient.

### 3.10 Conclusion

The mobile application serves as the user interface for the fall detection system. The screenshots showcased in this chapter provide a glimpse into the functionalities offered by the app. As development continues, we can expect further enhancements to the user experience, such as integration with wearable devices or more advanced data analysis features.

## Chapter 4: The Model

This chapter will detail the development and implementation of a custom You Only Look Once (YOLO) model for fall detection in computer vision.

### 4.1 Introduction

You Only Look Once (YOLO) is a powerful object detection framework known for its speed and accuracy. In our fall detection project, we leveraged a custom YOLO model. This model is trained to analyze entire images at once, allowing it to efficiently identify a person's pose and determine if they have fallen within the frame. This real-time detection capability is crucial for our project, as it enables us to quickly react to potential fall situations.

### 4.2 Data Collection

Building a robust fall detection model hinges on a diverse and informative dataset. To achieve this, we weren't limited to a single source. We collected images from:

- YouTube videos
- Instagram posts
- Roboflow

Additionally, we explored other image platforms that offered relevant content. This multi-source approach ensured our model encountered a wide range of fall scenarios, backgrounds, and lighting conditions, ultimately enhancing its ability to generalize and perform accurately in real-world situations.

We collected a total of 2400 images of people in falling motion.

**A significant amount of effort was invested in this step, as the quality of the data, directly influences the model's accuracy.**

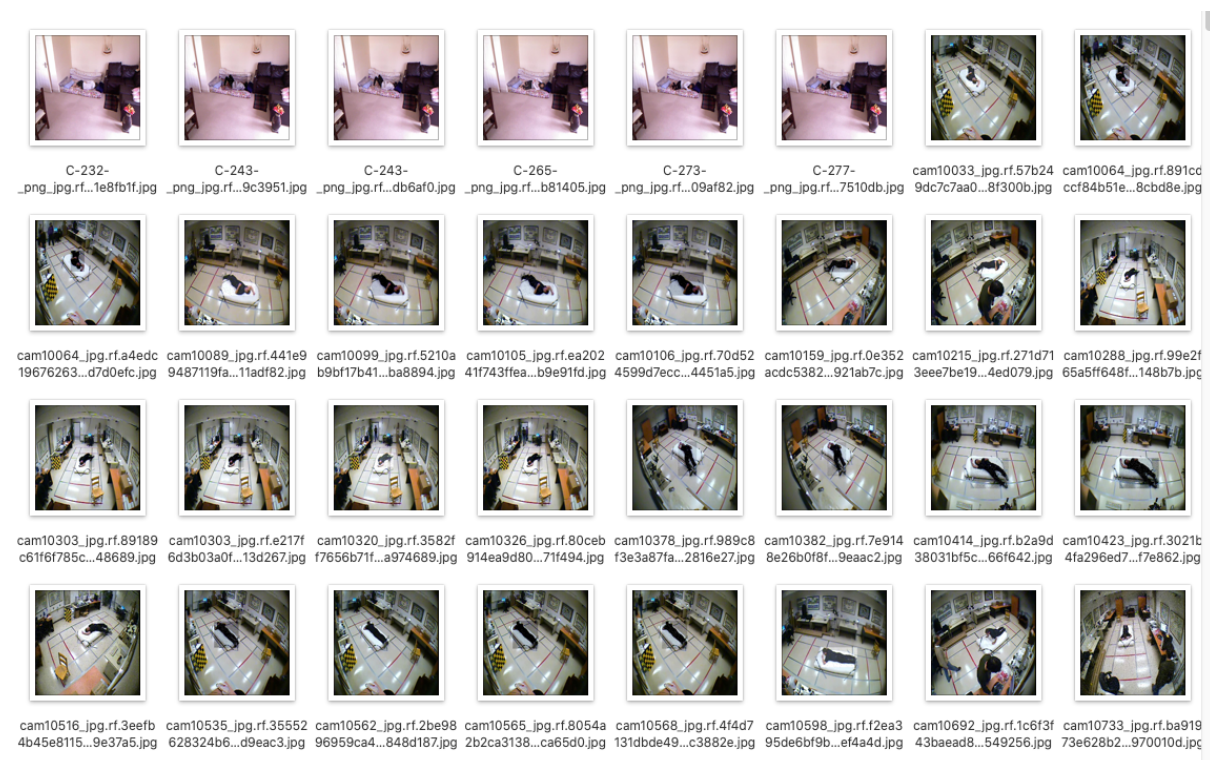


Figure 9: A screenshot showcasing a part of the dataset that was used to train the model

### 4.3 Data Annotation

After collecting the raw images, the next critical step involved meticulous annotation. This process entailed manually labeling each image to identify the presence or absence of a fall. We meticulously defined bounding boxes around individuals in the images, particularly focusing on poses indicative of falls.

This precise annotation provided the ground truth data for our YOLO model, allowing it to learn the visual characteristics associated with fall events. The quality and comprehensiveness of this annotation process directly influence the model's ability to accurately detect falls in unseen situations.

We used Roboflow to annotate the images.

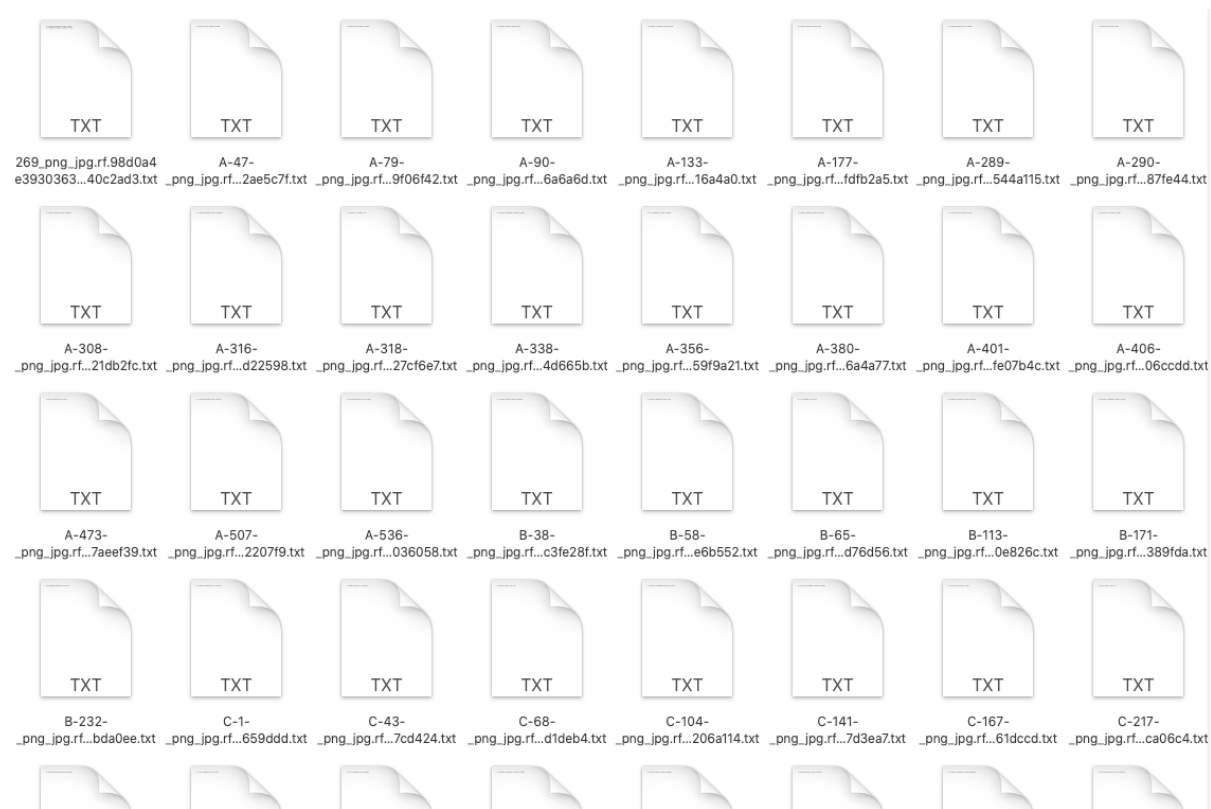


Figure 10: A screenshot showing the image annotation data

## 4.4 Preprocessing and Augmentation

Prior to training our YOLO model, we implemented a data pre-processing pipeline. This involved essential steps like resizing images to a consistent size and normalizing pixel values for better training efficiency. Initially, we experimented with data augmentation techniques like image rotation and flipping. However, during development of the first two model versions, we encountered a significant challenge. Augmenting images of fallen people by rotating or flipping them inadvertently caused them to resemble images of people standing. This confusion led to a high number of false positives in our model. To address this issue, the final version of the YOLO model relied minimally on augmentation and focused primarily on resizing and normalization. This approach minimized the risk of introducing misleading variations during training and ultimately led to improved fall detection accuracy.

## 4.5 Accuracy and Metrics

Our custom YOLO model achieved impressive results in fall detection. The model's mean Average Precision (mAP) reached 91.4%, demonstrating its overall effectiveness in correctly identifying falls across various scenarios. Furthermore, it achieved a high precision of 96.8%, indicating a low rate of false positives, meaning the model rarely flagged non-fall events as falls. While recall reached 81.4%, signifying the model successfully detected most actual falls in the test data. It's important to note that during our own internal testing with

unseen data featuring different lighting and angles, the model remarkably identified falls in all 10 test cases. This exceptional performance in a controlled setting highlights the model's potential for real-world application.

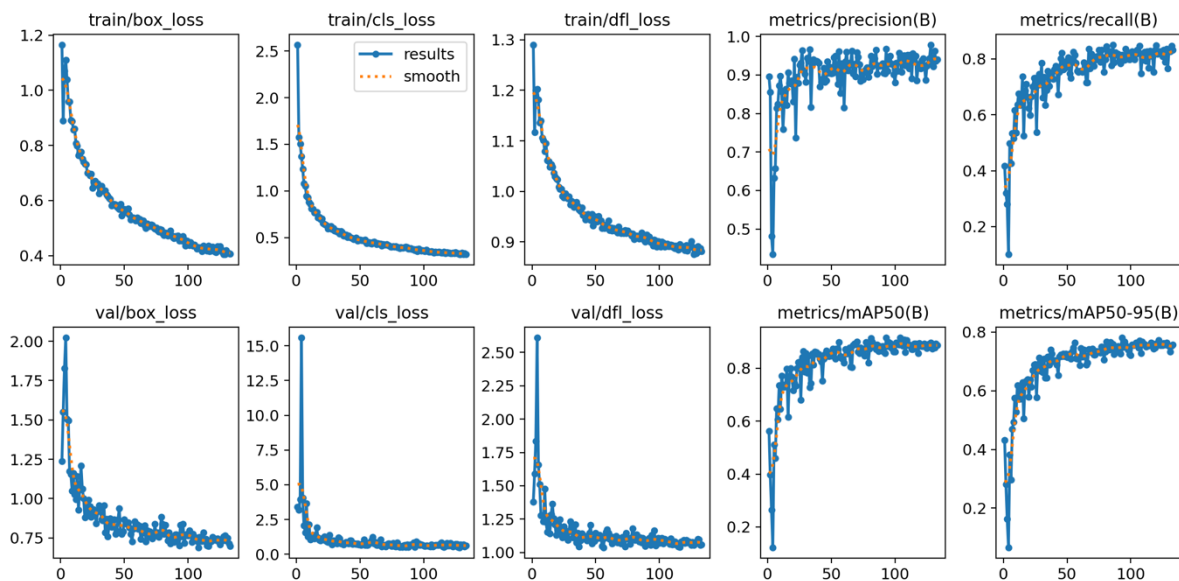


Figure 11: An image showing different metrics as our model trained

## Chapter 5: Connecting Intelligence to Action - Mobile App and Fall Detection Model

This chapter explores the critical link between the mobile application and the fall detection model running on the computer. This connection ensures timely notification and potential emergency response in the event of a fall being detected.

### 5.1. System Overview

The fall detection system comprises two key components:

**Computer-based Fall Detection Model:** This Python program analyzes video frames captured by a camera to identify potential falls.

**Mobile Application:** This user-friendly app installed on the resident's phone receives fall detection alerts and initiates pre-defined actions.

### 5.2. Communication Bridge: Firebase

Firebase, a Google Cloud platform, acts as the communication bridge between the two components. Here's how it facilitates data exchange:

**Python Script:** The fall detection model, upon detecting a fall, triggers a Python script.

**Data Transmission:** This script utilizes Firebase's functionalities to send a notification signal to the mobile application. The notification might include additional data like time of fall and resident information.

### 5.3. Mobile App in Action

The mobile application receives the notification from Firebase and initiates a pre-programmed sequence:

**Countdown Timer:** An on-screen countdown timer displays a configurable time window (timer is chosen by the user, default is 15s) allowing the resident to cancel the emergency call if the fall was a false alarm.

**Emergency Call:** If the countdown reaches zero without user intervention, the app automatically initiates a call to the pre-defined emergency contact number.

**SMS Notification:** Simultaneously, the app sends an SMS message containing the resident's location (obtained through GPS) to designated recipients, informing them of the potential fall incident.

### 5.4. Benefits of Real-Time Connection

This real-time connection between the mobile application and the fall detection model offers several advantages:

**Faster Response Time:** The system eliminates manual intervention, enabling faster emergency response in case of a fall.

**Improved Accuracy:** Firebase ensures reliable data transmission, minimizing the risk of missed notifications due to communication failure.

**Customizable Actions:** The mobile application allows for configuring emergency contacts, countdown timer duration, and SMS recipients, tailoring the response to individual needs.

### 5.5. Conclusion

The seamless connection between the mobile application and the fall detection model is the heart of the system's effectiveness. By leveraging Firebase for real-time communication, the system offers a prompt and potentially life-saving response in the event of a fall. This chapter has highlighted the crucial role this connection plays in enhancing the safety and well-being of the system's users.

## Chapter 6: Conclusion - A Safety Net for the Vulnerable

This document has explored the development and functionalities of the fall detection system. Throughout the chapters, we delved into the core components, from the computer-based fall detection model to the user-friendly mobile application. We also examined the

seamless communication facilitated by Firebase, ensuring prompt action in the event of a fall.

### 6.1 Impact and Future Considerations

The fall detection system offers a valuable safety net for individuals prone to falls, promoting independent living with peace of mind. Beyond individual users, the system can benefit caregivers, allowing them to remotely monitor the well-being of their loved ones

As technology continues to evolve, future iterations of the system can incorporate advancements in areas like:

- **Machine Learning:** Refining the fall detection model with a wider range of fall scenarios and activities can further enhance its accuracy.
- **Wearable Integration:** Integrating with wearable devices like smartwatches could offer additional data points for fall detection and improve user convenience.
- **Advanced Notifications:** Expanding notification options could include video clips of the fall event, providing valuable context to emergency responders.

### 6.2 A Final Note

The fall detection system serves as a technological solution aimed at promoting safety and independence. It is crucial to remember that the system functions best when used in conjunction with other safety measures like strategically placed grab bars and fall-proof flooring.

This documentation serves as a comprehensive guide to the system's functionalities, aiming to empower users and caregivers with knowledge and confidence. By fostering a proactive approach to fall prevention and offering a safety net in case of an incident, the fall detection system strives to enhance the quality of life for its users.