ECE 445

SENIOR DESIGN LABORATORY

FINAL REPORT

Final Report for ECE445: Intelligent Home Security System

Team #13

DANYU SUN
(danyu3@illinois.edu)
GE SHAO
(geshao2@2puksibd)
JIAXUAN ZHANG
(jz122@illinois.edu)
YUXIN XIE
(yuxinx7@illinois.edu)

Professor: Bruce Xinbo Yu
TA: Jingzhou Li

May 26, 2025

Abstract

As the population ages, an increasing number of the elderly cannot receive timely intervention after a fall. Delayed assistance may threaten their safety. Therefore, the project aims to develop an intelligent home monitoring and alert system that utilizes a mobile robot to monitor the activities of the elderly in real time. The key features of the system consist of autonomous trace, human detection, and fall motion detection. The robot carrying the whole system will adapt to various home environments to avoid obstacles, follow the elderly, and monitor the activities of the elderly. The system has several advantages, such as reducing response time to fall events and minimizing medical risks. Although the system is still facing risks and challenges, it will continue to update to implement more features, such as heart rate monitoring and voice interaction. In general, by mixing real-time monitoring and instant alerts, the system can help elderly people receive timely intervention, which can improve their safety and quality of life.

Contents

1 Introduction

1.1 Purpose and Problem Statement

With the changing social structure, the number of elderly people living alone is gradually increasing, making the issue of ensuring their personal safety in daily life an urgent problem. As is well known, falls are common among the elderly, and if they are not detected and addressed in time, they can lead to serious consequences or even become life-threatening. Therefore, it is important to develop an intelligent system with fall detection and immediate response.

Therefore, the aim of this project is to design and implement an intelligent robot with capabilities of human presence recognition, fall detection, automatic alarming, and autonomous following. The system first identifies whether a human is present in the camera view. If a person is detected, it will further determine whether the individual is in a falling state. Upon detecting a fall, the robot will immediately send out an alarm signal to alert the surroundings. At the same time, if the person is engaged in normal activity, the robot can autonomously adjust its position to continuously follow and monitor the individual.

By implementing this system, family members' concerns regarding the safety of elderly individuals living alone can be partially reduced. Moreover, it provides an intelligent and practical solution for home-based elderly care and remote monitoring scenarios.

1.2 Project Functionality Overview

The aim of this project is to develop an intelligent robot with human detection, fall recognition, and automatic alarm functions to improve the safety of elderly people living alone at home. The system realizes active monitoring of the elderly through the collaborative work of these three core functions: once a person is detected, the robot will continuously track his/her status, and once a fall occurs, the system will immediately send out an alarm, thus alerting the relevant personnel to intervene at the first time. This process forms a complete "detection-analysis-response" closed loop, effectively reducing the risk of accidents.

1.2.1 Human Detection

The module uses the YOLO image analysis algorithm to analyze the images captured by the camera in real time and determine whether an individual is present within the monitoring area. Once a person is detected, the robot will automatically activate subsequent function modules to further monitor the person's actions. Additionally, the robot is equipped with a distance sensor to measure the distance between the robot and the person. When the sensor detects that the person is far away, the robot will automatically move closer to the target to obtain a clearer image, thereby improving the accuracy and reliability of the monitoring.

Subsequent fall recognition and alarm response can only be realized if the system accurately detects the presence of a person. Therefore, the human detection function is a fundamental part of the system, ensuring that the robot always focuses on key objects and does not miss monitoring targets.

1.2.2 Fall Recognition

The fall recognition module is designed to determine whether a person in the monitoring screen has fallen. The system adopts the YOLO-NAS algorithm to analyze and recognize human posture in real time. Once a person is detected, fall detection is activated to monitor whether the posture rapidly changes from ”standing” to ”laying down”. If the likelihood of a fall exceeds a set threshold, it will be recognized as a fall, and the result will be immediately transmitted to the alarm module.

The fall detection module is the core component of the entire project. The overall goal of the project is to reduce the risk of unrecognized falls through active monitoring and a timely response by the robot. The module allows the robot to not only ”see” the person, but also ”judge” whether a dangerous situation has occurred. Once a fall is detected, an alarm is triggered immediately to ensure timely assistance. In this way, the system can effectively protect elderly people living alone, improving their sense of security and quality of life at home.

1.2.3 Automatic Alarm

When the fall detection module detects a fall, the automatic alarm module will immediately activate, triggering the buzzer to emit a loud alarm sound. In this way, the robot is able to alert people around it in time at the scene, attracting attention and prompting others to come to help as soon as possible.

The automatic alarm module ensures that falls can be quickly detected and reported, effectively reducing the time an elderly person remains unattended after a fall and helping to get assistance as early as possible, thus enhancing the safety and quality of life of elderly people living alone.

1.3 Subsystem Overview

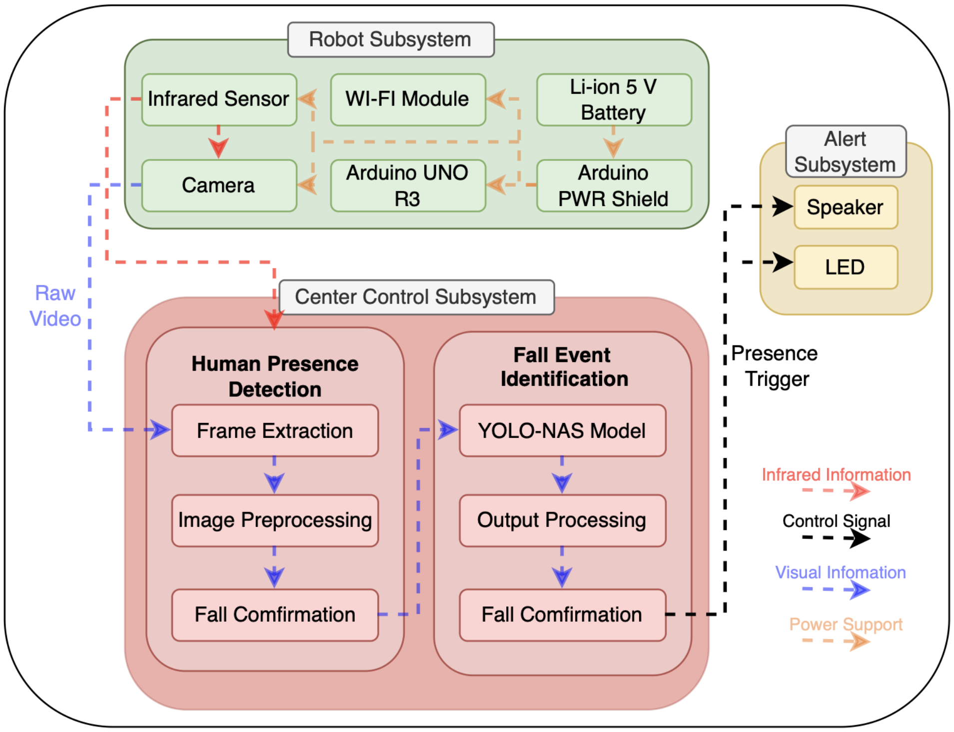

Figure 1: Block Diagram

The overall system of this project consists of three subsystems: robot subsystem, center control subsystem, and alarm subsystem, as shown in Figure 1. Each subsystem is responsible for distinct functional tasks, and they work collaboratively to accomplish the complete process of human detection, fall recognition, and alarm response. The robot subsystem is mainly responsible for collecting visual and distance information from the environment and transmitting video data to the center control subsystem in real time. The central control subsystem processes and recognizes the received images to determine whether a person is present and whether a fall has occurred. If a fall is recognized, the system will activate the alarm subsystem via the control signal to issue a clear audible alert. The subsystems are interconnected through serial communication and control logic, ensuring stable and efficient data transmission and coordinated operation.

1.3.1 Robot Subsystem

The robot subsystem is built around an embedded development board and is equipped with modules such as a camera, infrared sensors, and motors. It handles data acquisition and local behavioral responses. The system is capable of capturing real-time video images of the surrounding environment and steadily transmitting the images to the central control unit for further processing. Meanwhile, the ultrasonic sensor is used to measure the relative distance between the robot and the human body or obstacle in front of it. If a human is detected at a distance, the robot will activate the motion control mechanism to automatically approach the target to ensure the clarity of the captured images. This subsystem maintains the availability of image and distance data under a variety of environmental conditions, which is the prerequisite for the normal operation of subsequent recognition functions.

1.3.2 Center Control Subsystem

The central control subsystem, as the core of image recognition and analysis, is responsible for processing the video input and determining human behavior. The module first performs frame sampling and image preprocessing on the video stream transmitted from the robot subsystem. Then, the YOLO-NAS [1] model is utilized to detect the presence of a human target in the image. After confirming the presence of a person, the system further uses the YOLO-NAS model to analyze the human body posture and identify whether a fall has occurred. The system will determine whether it is a real fall event based on the posture change. Once the conditions are met, the system immediately sends out the control signal to activate the alarm subsystem. The accuracy and real-time performance of this module directly affect the response efficiency and safety of the overall system, making it a critical component in achieving reliable fall detection.

1.3.3 Alert Subsystem

The alarm subsystem is the execution unit that executes the response in this system. When the central control subsystem sends out a fall confirmation signal, the alarm module will activate the buzzer to emit a sound alert to notify the surrounding personnel. The alarm is triggered with minimal delay, enabling a near-instantaneous audible response after a fall is detected.The sound coverage is sufficient to deal with the typical home environment. This module is an important bridge from fall detection to actual intervention.

2 Design

2.1 Subsystem Implementation

2.1.1 Robot Subsystem Hardware

As shown in Figure 2, all components are housed on a copper-clad backplane, with a 7.4 V Li-ion battery pack powering the Arduino UNO, the power driver PCB, and the wireless communication PCB. The circuit shown in Figure 3 is designed to connect the battery and other key components. The Arduino and the power driver board work together to control several functional modules: four drive motors to help the cart move forward and backward, two servo motors to control the gimbal in two degrees of freedom, infrared sensors and ultrasonic sensors to control the cart to follow a person, and an IR sensor and ultrasonic sensors to control the cart to follow a person. Two servo motors can control the rotation of the gimbal in two degrees of freedom, and infrared and ultrasonic sensors can control the cart to follow people. The camera module is connected to the wireless communication PCB to capture images and transmit them to the computer for data processing.

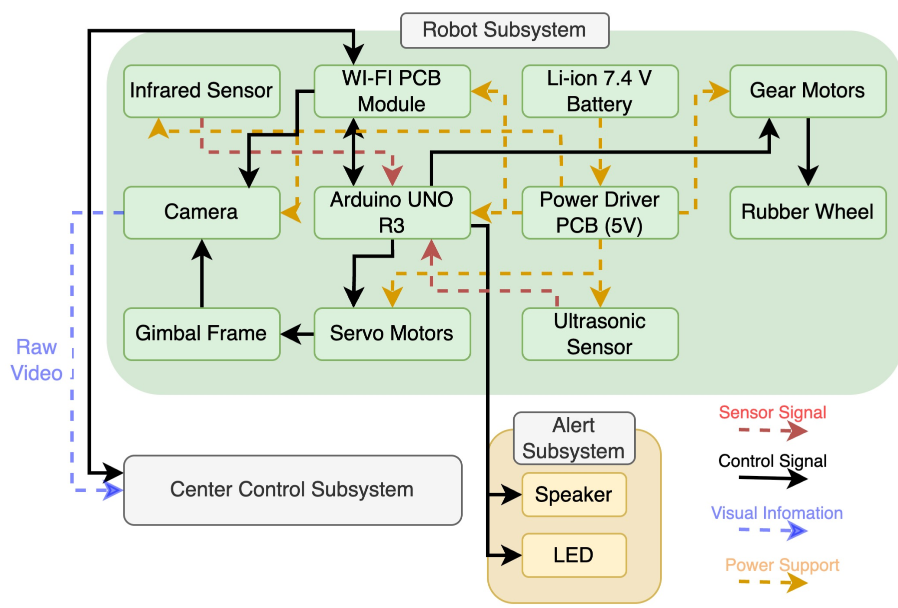

Figure 2: Robot Subsystem and Alert Subsystem Diagram

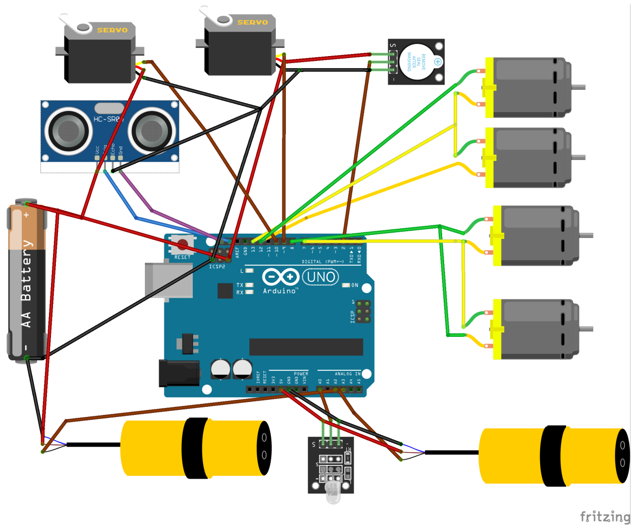

Figure 3: Robot Circuit Schematic

Here are details of key components:

Li-ion Battery: A 7.4 V Lithium-ion battery pack with dimensions of 67 mm by 37 mm can connect to the Arduino UNO via the DC barrel jack to power the motors and electronics.

Multifunctional Baseboard: The baseboard is made of copper-clad material, with dimensions of 160 mm by 255 mm, and is used to connect the motors, a printed circuit board, and gimbal.

USB Video Camera: The camera is connected with the wireless communication PCB via USB to capture 640 × 480 pixel images, and transmit them to the computer. It measures 62 mm × 50 mm × 66 mm and is fixed on the pan–tilt gimbal.

TT DC Gear Motors: Four motors are connected to the Arduino UNO and wheels. Each motor can operate at 3 – 6 V with a 1:48 gearbox ratio, ensuring smooth and stable wheel propulsion.

Rubber Wheels: The wheels are made of rubber with a 65 mm diameter and a 10 mm D-shaft, providing enhanced traction and stability for the robot.

Two-DOF Gimbal Frame: The frame can hold two 9 g servos and provides 180° rotation across two degrees of freedom, enabling the camera to capture video more effectively.

D80K Infrared Sensors: Sensors use infrared emission and reflection to measure the distance between the robot and obstacles within 80 cm. By analyzing changes in distance over time, it can track a person's movement trajectory.

HC-SR04 Ultrasonic Sensor: The sensor can measure distances from 2 cm to 450 cm using ultrasonic pulse time-of-flight; by analyzing changes of measured distance over time, it tracks the human's movement trajectory.

Nine-Gram Servo Motors: Motors can provide sufficient torque to enable 180° rotation across two degrees of freedom for the pan–tilt gimbal.

Wireless Communication PCB: The PCB allows Bluetooth and Wi-Fi transceivers to connect to the Arduino via UART and USB for real-time data exchange and video streaming.

Power Driver PCB: The PCB can regulate input voltage, efficiently distribute current, and interconnect all system modules.

Microcontroller PCB (Arduino UNO R3): The microcontroller serves as the central processing unit. It can receive input signals from different sensors and relay signals to the wireless communication module. It can also execute algorithms to control the wheel motors, and output signals to trigger the alert subsystem.

2.1.2 Alert Subsystem Hardware

As shown in Figure 2, a 10 mm LED and a YXDZ active buzzer are connected to the power driver PCB for fall notifications.

Here are details of key components:

10 mm LED Module: A high-intensity 10 mm LED module with a built-in current-limiting resistor, operating at 5 V and drawing approximately 20 mA, is mounted on the alert PCB for clear visual notification upon fall detection.

YXDZ active Buzzer: An active piezoelectric buzzer from YXDZ Youyuan, operating at 5 V and drawing 30 mA, providing an 85 dB tone at 2.5 kHz when activated by the Arduino digital output.

2.1.3 Control Subsystem Software

This control subsystem software, as shown in Figure 4, is deployed in the video processing computer connected to the robot's camera. It is responsible for accomplishing the control tasks of image acquisition, fall detection, and alarm commands. It is also the core part of the whole system for realizing intelligent recognition and decision-making.

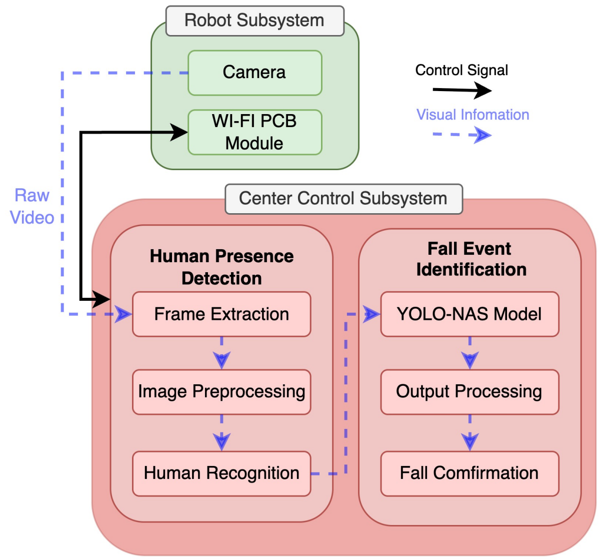

Figure 4: Center Control Subsystem Diagram

Image Capture Module: The module connects to the camera module through the USB interface and continuously receives video streams with 640 x 480 resolution. The sampling frame rate is adjustable, and the default setting is 15 frames per second to balance the image quality and real-time processing efficiency[2]. The captured images are passed into the "Human Presence Detection" sub-module, where frame extraction and image preprocessing are completed sequentially to provide standardized inputs for subsequent detection tasks.

Fall Detection Algorithm: In this project, a trained YOLO-NAS model is used to detect human poses for each image frame and be combined with temporal feature analysis to realize the fall determination. The model is trained on a public fall dataset, and the validation accuracy exceeds 80%, and the false alarm rate is controlled within 5%.

Event Recognition and Decision Making Module: When the algorithm determines that a fall event has been detected, the system immediately sends an alert command to the Arduino main control board through the serial port, triggering the LED and buzzer alarms.

2.2 Equations and Simulations

2.2.1 Hardware

To test the verification of using ultrasonic sensors to avoid obstacles in this project, an interface is created that connects the Arduino UNO R3 and the HC-SR04 ultrasonic sensor using Tinkercad, as Figure 5 shows.

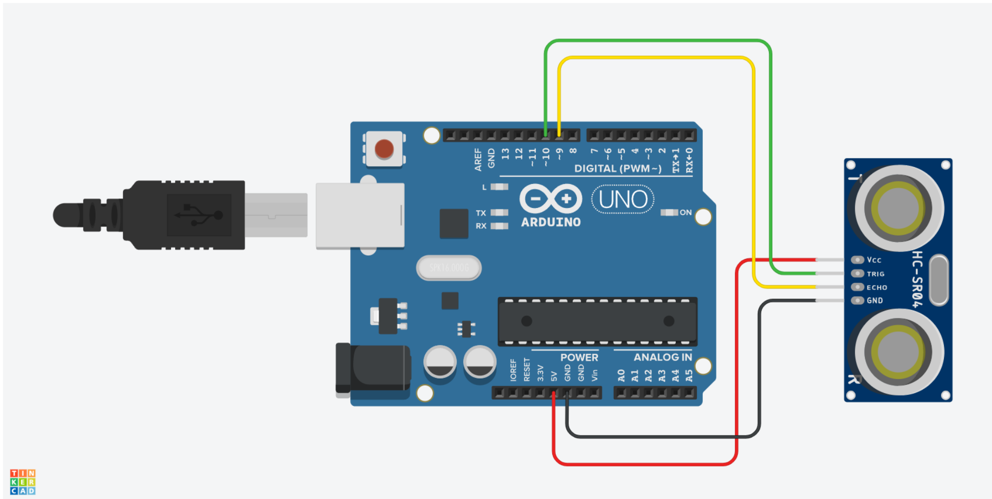

Figure 5: Interface of Ultrasonic Sensor and Arduino using Tinkercad

Figure 6 presents a simulation interface of the HS-SR04 sensor that is performed using the C++ code.

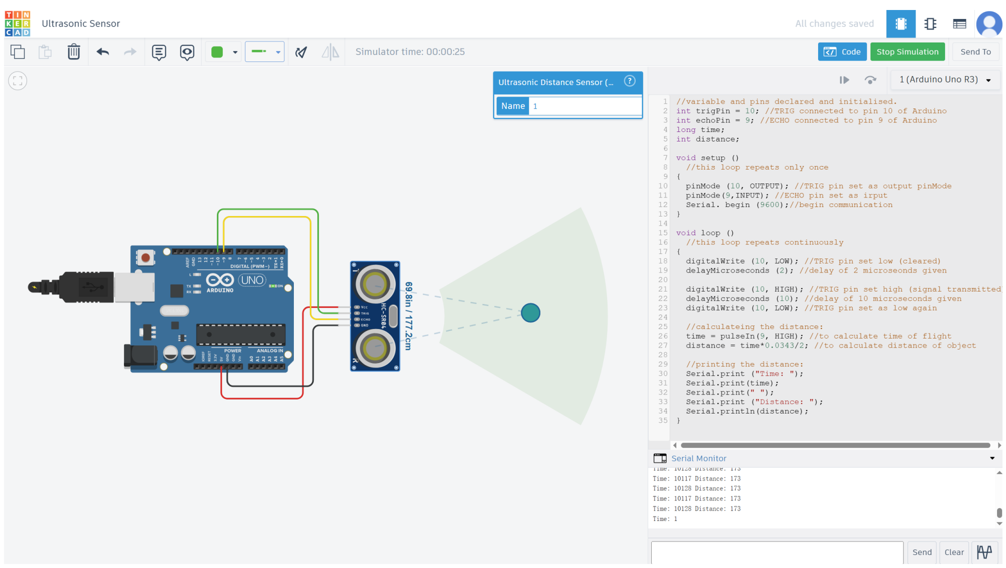

Figure 6: Screenshot of the Ultrasonic Sensor Simulation Interface

From the Serial Monitor in the interface, the output times and actual distances are recorded. Based on the recorded times, the corresponding distances are calculated, respectively, using Equation (2.1).

$$\text{Distance} = \frac{v \cdot t}{2} \tag{2.1}$$

Where:
v = speed of sound in air, approximately 343 m/s

t = round-trip time measured by the ultrasonic sensor
The division by 2 accounts for the go-and-return travel of the sound wave.

Comparing the actual distances and the calculated distances, the errors for the sensor's distance measurement can be calculated as shown in Table 1.

Table 1: Simulation Data for the HC-SR04 Ultrasonic Distance Sensor

Time (microseconds)	**Calculated Distance (cm)**	**Actual Distance (cm)**	**Error (%)**
2260	38.76	39.4	1.623
2929	50.23	51.3	2.081
3520	60.37	61.6	2.00
4623	79.28	80.9	2.00
5482	94.02	95.8	1.86
6494	111.37	113.5	1.87
7510	128.80	131.3	1.91
8524	146.19	149.1	1.95
10128	173.70	177.2	1.98

As Table 1 shows, the error between the calculated distance (the distance that the ultrasonic sensor measures) is around 2%, which is quite low and verifies that the error can be ignored when the robot is doing obstacle avoidance and human tracking tasks.

To achieve the function of human tracking, another important hardware component is the infrared sensor. Infrared (IR) sensors are commonly used for short-range obstacle detection due to their fast response and low cost. In this project, multiple IR sensors are strategically placed to expand the detection coverage in front of the robot. The detection coverage is a sector that determines how much area in front of the robot it can "see", and it can be calculated using Equation (2.2).

$$A = \frac{1}{2}R^2\theta \tag{2.2}$$

Where:
A — infrared sensor coverage area (in cm^2)
R — maximum detection range (in cm)
θ — sensor field of view (FoV, in degrees or radians)

Figure 7 shows the 2D visualization of an E18-D80NK IR sensor's field of view (FoV = 15°, Range = 80 cm). The shaded area indicates the detection coverage zone. Obstacles outside this area may not be detected, leading to potential blind spots. This supports sensor placement decisions.

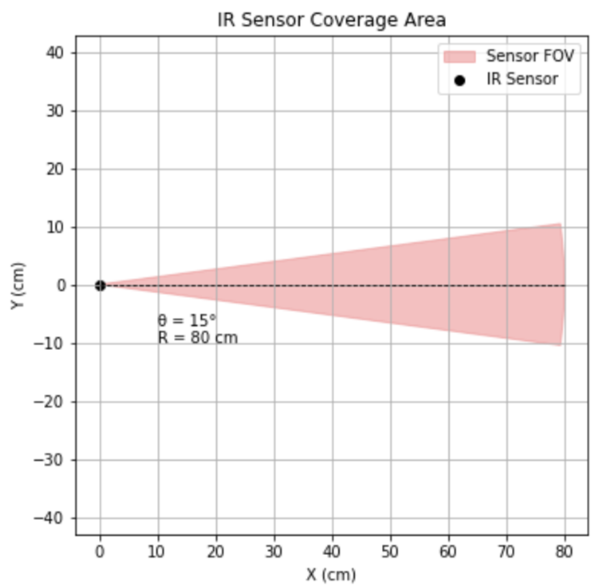

Figure 7: 2D Field of View of an E18-D80NK IR Sensor

Figure 8 shows the field of view simulation for three E18-D80NK sensors mounted at -30°, 0°, and +30° relative to the forward direction. Each sensor has a field of view of 15° and a range of 80 cm. The combined coverage helps reduce blind spots directly in front of the robot.

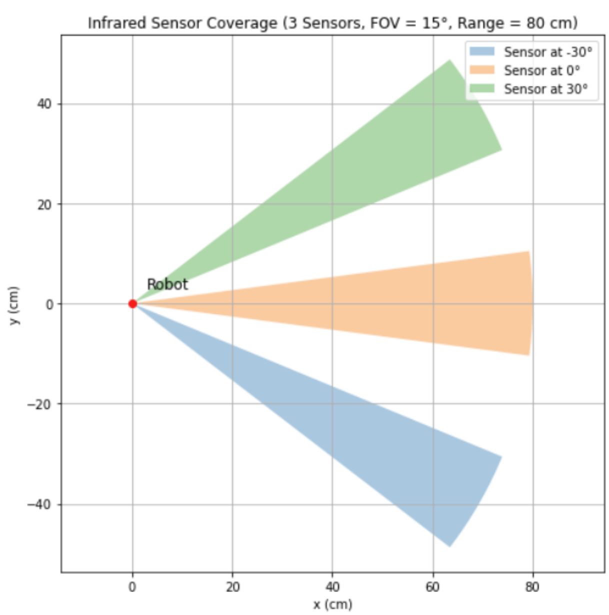

Figure 8: 2D Field of View of three E18-D80NK IR Sensors

Considering that human movement tends to be continuous and within a limited space (e.g., a hallway or a room), this sensor placement ensures that the robot can maintain perception of the target as long as the human stays within 0.8 m ahead and ±30° laterally. This design not only effectively reduces blind spots but also supports the principle of maintaining a safe 0.5–1 meter distance between humans and robots in indoor settings.

2.2.2 Software

In the software design, the fall detection algorithm module adopts the currently advanced target detection algorithm YOLO-NAS. The algorithm maintains high detection accuracy and has a fast inference speed, which is suitable for embedded systems with high real-time requirements.

To evaluate the performance of the model, we conducted simulation experiments on publicly available fall behavior datasets, such as the UR Fall Detection Dataset[3]. During the testing process, we quantitatively analyzed the model output by selecting a number of commonly used performance metrics, including precision, recall, and F1 score[4], which are defined as follows:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{1}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{2}$$

$$\text{F1 Score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{3}$$

Where TP denotes the number of samples in which real falls are correctly recognized, TN is the number of samples in which non-falls were correctly recognized, FP is the non-fall samples that were misdiagnosed as falls, and FN is the fall samples that were missed.

TEST RESULTS

Metric	Value
Precision@0.50	0.846154
Recall@0.50	0.76393
mAP@0.50	0.801402
F1@0.50	0.790094
Best_score_threshold	0.53

Figure 9: Simulation Result of Fall Detection

The experimental results in Figure 9 show that our system achieves 84.7% Precision, 76.4% Recall, and 79.0% F1 Score on the fall detection task. The results show that the YOLO-

NAS model has good stability and reliability in detecting fall behaviors, which meets the design requirements of the fall recognition function in this project.

In conclusion, the selection of YOLO-NAS is not only theoretically supported, but also verified by data testing, with clear performance advantages, which reasonably supports the overall software function module design of this system.

2.3 Design Alternative Analysis

Several design issues appeared during the design of infrared sensors for developing the human-following feature. Firstly, the two parallel infrared sensors in the front couldn't accurately detect whether a person was moving left or right. To solve this problem, we repositioned the two infrared sensors with a larger angle (around 60 degrees) between them. In this way, a larger field of view was obtained to increase orientation awareness, but the problem still exists due to the inherent range limitations of the infrared sensors. Secondly, the infrared sensors couldn't distinguish between different types of moving objects. For example, they were unable to tell whether a detected obstacle was a person or a pet. To partially solve this, two solutions were discussed. One was using more integrated cameras with human detection functions. Another one was replacing the Arduino UNO with Raspberry Pi, which had a stronger processing capability. However, due to the limit of budget, both suggestions couldn't be adopted.

Another key issue was the sensitivity of infrared sensors to environmental disturbances. Sensor readings became unstable under high ambient light conditions, especially under direct sunlight near windows. In one test scenario, false negatives increased by about 40% when a person wearing black clothing walked through a sunlit area. This was because both bright and dark absorbing surfaces reduced the reflectivity of the infrared signal. Therefore, we limited our testing and intended use to controlled indoor environments with consistent illumination. However, the degraded performance of variable illumination remained a limiting factor.

A further design issue arose during the development of the fall recognition module. Initially, the algorithms used for image processing resulted in slow screenshot capture, leading to significant delays of over 10 seconds in frame analysis and making decisions. This delay significantly affected the system's ability to respond to real-time visual cues. To address this problem, we implemented an alternative algorithm and adjusted the screenshot interval to 2 seconds through multiple rounds of testing and optimization. This modification successfully reduced the processing delay to less than 200 ms.

Another design trade-off involved the selection of the LED for the alert subsystem. A high-luminance LED was initially considered to ensure visibility under bright lighting conditions, but it came with higher power consumption and required current-limiting resistors with stricter specifications. We ultimately chose a standard-brightness LED, which provided sufficient visibility in indoor environments while maintaining low power consumption and simplifying the circuit design.

3 Cost and Schedule

3.1 Bill of Materials

Table 2 presents a detailed breakdown of all hardware components used in the system, including their specifications, quantities, manufacturers, and unit prices. This bill of materials serves as the cost baseline for the project, with a total estimated expense of $26.62. The selected components balance affordability and performance, supporting key functions such as sensing, computation, communication, and actuation.

Table 2: Bill of Materials

Component Name	Value	Quantity	Manufacturer	Price (USD)
Arduino UNO R3	ATmega328P, USB, 16 MHz	1	YwRobot Studio	21.95
Motor Driver Module	L298N, Dual H-Bridge, 2 A/channel	1	YwRobot Studio	3.71
WiFi Video Module	802.11 b/g/n, USB2.0, 32 MB RAM, 8 MB Flash	1	XiaoR Geek	28.45
Camera, USB	640 x 480, USB, 5 V, 62 x 50 x 66 mm	1	XiaoR Geek	12.50
Chassis PCB	Power + Motor pads, 2-layer PCB	1	Zave	9.00
DC Gear Motor, TT	3–6 V, 1:48 gearbox, 200 mA @ 6 V	4	XiaoR Geek	7.20
Wheel, Rubber	65 mm diameter, D-shaft, rubber tire	4	Zave	0.60
Gimbal Frame, 2-DOF	Pan/Tilt, servo mount compatible	1	XiaoR Geek	1.80
Battery Pack, Li-ion	7.4 V, 2200 mAh, 8 A protected	1	Chenke store	5.50
Charger, Li-ion	AC 110–240 V, DC 12.6 V/0.5 A	1	Chenke store	4.00
USB Cable, A-B	USB-A to USB-B, 360 mm	1	XiaoR Geek	0.50
IR Line Sensor	IR, 3.3–5 V, analog+digital out	2	XiaoR Geek	0.90
Ultrasonic Sensor	HC-SR04, 5 V, 2–450 cm	1	XiaoR Geek	0.80
Total Cost (USD)				**121.21**

3.2 Project Timeline

This section outlines the development schedule of the project throughout the semester. Table 3 summarizes weekly tasks assigned to each team member, providing a clear view of individual responsibilities and collaborative progress. The work plan follows a logical sequence from design and procurement to integration, testing, and final presentation, ensuring timely delivery of all milestones.

Table 3: Weekly Schedule by Member

Week	Ge Shao	Danyu Sun	Jiaxuan Zhang	Yuxin Xie
2/17	Review fall detection requirements; plan image acquisition module	Research YOLO-NAS model structure; plan inference setup	Review chassis designs and finalize component selection	Study tracking logic and review IR distance sensing methods
2/24	Write draft design document; prepare for camera testing	Confirm YOLO-NAS pretrained weights; set up testing codebase	Assemble base frame of the robot chassis	Prepare IR sensor module wiring and test simple outputs
3/2	Purchase camera, LED, buzzer, and connect to PC	Write image inference script for model input/output	Install servo system and verify camera mounting stability	Test IR distance feedback logic and debug noise issues
3/9	Implement camera data streaming and verify frame rate	Test model output on sample video and label accuracy	Finish physical mounting of sensors and servos	Develop PID-like logic for following behavior
3/16	Integrate video stream with PC-based processing	Assist in detection debugging and compare outputs	Begin initial vehicle movement tests (forward/turn)	Link IR data to motor logic for basic follow demo
3/23	Set up continuous video acquisition pipeline	Deploy YOLO-NAS and run first fall detection test	Install buzzer and LED modules	Assist in motion trigger signal wiring
3/30	Debug person detection flow and mark valid areas	Fine-tune the YOLO-NAS model	Fix mechanical bugs in wheels and servo control	Validate autonomous forward/rotation behavior
4/6	Connect detection result to alarm trigger module	Test alarm logic flow with real-time fall input	Tune mechanical structure for sensor accuracy	Stabilize following behavior with smoother feedback

Week	Danyu Sun	Ge Shao	Jiaxuan Zhang	Yuxin Xie
4/13	Test complete: video → detection → alarm chain	Help simulate multiple fall scenarios and measure latency	Perform integration of movement with detection	Test follow behavior with dynamic targets
4/20	Write code for image-to-fall-alert logic loop	Benchmark detection speed and reduce false positives	Fix servo drift; secure wiring layout	Test system on real terrain with obstacle presence
4/27	Co-write final report and summarize detection results	Prepare final YOLO-NAS documentation	Refactor mechanical design and verify mobility	Evaluate follow precision in larger space
5/4	Record camera footage for demo	Collect logs of detection-to-alarm cases	Demo robot movement with detection system	Tune IR logic and record follow behavior for video
5/11	Participate in final demonstration and presentation	Participate in final demonstration and presentation	Participate in final demonstration and presentation	Participate in final demonstration and presentation

4 Requirements and Verification

4.1 Completeness of Requirements

For the robot subsystem, requirements are: The camera must deliver video with minimal delay (200 ms) to ensure real-time tracking of moving subjects. The infrared sensor must accurately detect human movement to enable effective following within a set range (1 m).

For the center control subsystem, requirements are: The system must process video input in real-time with a latency of less than 200 ms. The human tracking and recognition accuracy must exceed 90% under normal indoor lighting. The detection accuracy of fall accidents must exceed 80% under normal indoor lighting.

For the alert subsystem, the requirements are the following: The LED must be activated within 1 second after the system detects a fall. The speaker must sound an alarm within 1 second of receiving a fall signal and continue for over 10 seconds.

For all the above requirements, we have conducted experiments to verify them, and the results show that we have fully achieved all of them.

4.2 Verification Methods and Test Results

Requirement 1: Low-latency video transmission

Verification: A subject performed predefined motions while the video stream was logged. Timestamps between real-world action and display output were compared. 30 trials were conducted.
Result: Average latency was **148 ms**, maximum **172 ms**, both within the <200 ms threshold.

Requirement 2: Accurate human movement detection

Verification: A person walked in front of the robot at 0.5 m, 1 m, and 1.5 m, at speeds of 0.5 m/s, 1.0 m/s, and 1.5 m/s. For each condition, 10 tests were recorded to determine whether the IR sensor were triggered correct tracking behavior.
Result: The success rate of the experiment is shown in Table 4.

Table 4: Detection success rates at different distances and speeds

Distance (m)	Speed (m/s)	Success Rate
0.5	0.5	100%
0.5	1.0	100%
1.0	1.0	100%
1.5	1.0	90%
1.5	1.5	70%

IR detection is reliable within 1 m at all tested speeds.

Requirement 3: Real-time video input processing (latency <200 ms)

Verification: Live video was streamed to the system while a subject performed timed gestures. Timestamps were recorded at the moment of physical action and upon system recognition. 30

trials were conducted under consistent indoor lighting.
Result: The average processing latency was **132 ms**, with all results below **175 ms**, satisfying the <200 ms real-time requirement.

Requirement 4: Human recognition accuracy over 90% under indoor lighting

Verification: Subjects wearing common clothing colors (black, white, gray, blue, red) were tested against three background colors (white wall, brown wood, green curtain). Each combination was tested 5 times while subjects walked or stood. A total of 75 trials were manually labeled and compared with system outputs.

Result: Table 5 presents the results of human recognition under different situations.

Table 5: Human recognition accuracy for different clothing and background color

Background	Clothing Color	Detection Accuracy
White	Black	100%
White	Gray	100%
White	Blue	100%
White	Red	100%
White	White	80%
Brown	Black	100%
Brown	Gray	100%
Brown	Blue	100%
Brown	Red	100%
Brown	White	96%
Green	Black	100%
Green	Gray	100%
Green	Blue	100%
Green	Red	100%
Green	White	92%

Total tests: **75** Overall average accuracy: **97.6%**

Minor accuracy drops occurred when white clothing blended with light-colored backgrounds, but overall recognition performance exceeded the requirement threshold.

Requirement 5: Fall detection accuracy over 80% under indoor lighting

Verification: Subjects wearing common clothing colors (black, red, white) were tested against two background colors (white wall, brown wood). The subjects respectively used the forward fall, backward fall, and lateral fall (facing and back to camera) methods, each for 5 times. A total of 120 trials were manually labeled and compared with system outputs.

Result: Table 6 shows the impact of different backgrounds, clothing colors and falling postures on the accuracy of fall detection. Overall, the greater the difference between the color of the clothes and the background color, the higher the detection accuracy will be. Additionally, when the back of the body is facing away from the camera during the check, the accuracy will decrease.

Table 6: Fall Detection Accuracy under Various Background and Clothing Colors

Background	Clothing Color	Fall Posture	Detection Accuracy
White	Black	Forward Fall	100%
		Backward Fall	100%
		Lateral Fall (facing)	100%
		Lateral Fall (back to)	80%
White	Red	Forward Fall	100%
		Backward Fall	100%
		Lateral Fall (facing)	100%
		Lateral Fall (back to)	80%
White	White	Forward Fall	80%
		Backward Fall	60%
		Lateral Fall (facing)	60%
		Lateral Fall (back to)	40%
Brown	Black	Forward Fall	100%
		Backward Fall	80%
		Lateral Fall (facing)	100%
		Lateral Fall (back to)	60%
Brown	Red	Forward Fall	100%
		Backward Fall	80%
		Lateral Fall (facing)	80%
		Lateral Fall (back to)	60%
Brown	White	Forward Fall	100%
		Backward Fall	100%
		Lateral Fall (facing)	100%
		Lateral Fall (back to)	80%

Total tests: **120** Overall average accuracy: **85.0%**

Minor accuracy drops occurred when white clothing blended with light-colored backgrounds, or when subjects turned their backs to the camera. But overall recognition performance exceeds the requirement threshold.

Requirement 6: Quick response of the alert subsystem

Verification: Fall events were simulated, and the activation time of both the LED and speaker, and the duration of the speaker alarm, were measured using a stopwatch. Each test was repeated 10 times under consistent indoor conditions.

Results: The LED activated in an average of 0.52 seconds (range: 0.41–0.62 s), and the speaker in 0.51 seconds (range: 0.47–0.55 s), with the alarm lasting an average of 10.7 seconds (range: 10.3–11.1 s). Both components met the requirements (activation time <1 s, continue >10 s).

5 Conclusion

5.1 Accomplishments

The automatic alarm module realizes the function of rapid detection and automatic alarm for fall events of the elderly. The system is able to send out an alarm within a few seconds after a fall occurs, shortening the unnoticed time. Through several experimental tests, the accuracy of fall detection reaches about 90% with a low false alarm rate. The module has been tested in multiple rounds in a simulated environment with stable and responsive performance. Overall, the project achieves the expected core functions, provides effective support for the safety and security of independent living for the elderly, and has the potential for further optimization and practical application.

5.2 Uncertainties

During the test, the system is able to accurately trigger the buzzer to sound an alarm within 0.2 seconds through the infrared sensor when the distance of the obstacle is less than 15cm, and the response speed is satisfactory. However, the automatic obstacle avoidance function is still not realized, and the motors are unable to make timely stop or go-around actions according to the sensor inputs, resulting in the robot still hitting the obstacles in front of it in many experiments. In a total of 10 experimental trials, successful obstacle avoidance occurred in only 2 cases, yielding a success rate of 20%. This result highlights significant limitations in the current implementation of linkage control between sensors and the motion control components. Currently, the infrared signal only triggers the buzzer and fails to synchronize the motor control commands. In addition, the system uses a polling mechanism to process the sensor data, which has limited response speed and makes it difficult to avoid obstacles especially under concurrent workloads. To address this limitation, we propose a multi-faceted approach. First, an interrupt-driven mechanism is implemented to replace the polling method, significantly reducing system latency and improving responsiveness. Second, we integrate a state-based logic into the main control loop to prioritize obstacle detection signals. This ensures that critical safety-related interrupts are processed with minimal delay. These improvements will significantly enhance the practicality and robustness of the system.

5.3 Future Work

To enhance the autonomy and practicality of our fall detection robot, we identify three critical directions for future development.

5.3.1 SMS Notification System

Currently, the Arduino UNO R3 communicates wirelessly with the computer via the WiFi video module. We can utilize this existing communication link to extend the SMS notification function without adding additional GSM hardware. Specifically, when the YOLO-NAS model running on the computer detects a fall event, the computer can call the SMS gateway API (e.g., Twilio or AliCloud SMS service) via Python script to send an HTTP POST request containing the emergency message and the recipient's number. This solution makes full use of the existing network connection, requires no additional hardware cost, and can quickly send the alarm message to the family members in an unattended situation, improving the practicality and response efficiency of the system.

5.3.2 Temporal Fall Detection

The current detection process uses fixed interval video frame captures for image-based inference. The flow can be further extended by introducing a frame buffering mechanism (e.g., storing consecutive frames from the past 1 to 2 seconds) to utilize temporal motion features. These frame sequences can be passed as input to lightweight action recognition model such as MoveNet Multipose combined with LSTM post-processing, or a customized 1D Convolutional Neural Network (1D CNN) for motion vector classification. These models can be run on the PC, combined with OpenCV and TensorFlow Lite implementations. The approach significantly improves robustness to the detection of complex situations, such as slow falls, while maintaining a lower computational load and high processing speed compared to models using full videos.

5.3.3 Autonomous Charging System

Currently, 7.4V 2200mAh Li-ion batteries must be charged manually through a 12.6V/0.5A charger. To eliminate human intervention, an automatic docking charging mechanism is proposed. The existing infrared line sensor (3.3-5V) can be repurposed for simple path-following behavior towards the charging cradle. When a low voltage is detected (measurable by a voltage divider circuit connected to an Arduino analog pin), the robot will autonomously navigate to the charging cradle. The charging interface can be aligned using sheet metal contacts and spring-loaded jumper pins mounted on the bottom of the chassis, via a V-shaped funnel-style docking station. Internally, a charge controller (e.g. TP4056 module with 8A fuse for protection) can be embedded to ensure safe automatic charging without risk of overcurrent. When full charging is detected, an auxiliary microcontroller or timing circuit can cut off the power supply to protect the Li-ion battery cells.

5.4 Ethical Considerations

This project strictly adheres to the IEEE Code of Ethics [5], addressing privacy, safety, integrity, and social responsibility. Risk mitigation strategies and compliance with relevant standards are integrated into every stage of the system design and testing process.

5.4.1 Concern about Privacy

Our project involves the development of fall detection algorithms, which require video data for training and testing. However, the collection and use of human activity data raise privacy concerns, as improper handling of sensitive video footage may expose an individual's personal information. To ensure individual privacy, we will use publicly available fall detection video datasets to train and test our recognition algorithms. No personally identifiable or sensitive data will be collected, stored, or used in our research.

5.4.2 Concern about Public Understanding

Intelligent systems, especially those related to health and safety, can significantly impact individuals and society. However, a lack of clear understanding of these technologies can lead to misinterpretation, misuse, and even mistrust of automation. To avoid these risks, we will ensure that the design principles and instructions on how to use our systems are presented in an accessible way. We will also fully explain the features, limitations, and potential risks to help users fully understand our intelligent systems.

5.4.3 Concern about Integrity and Accountability

Developing a reliable and accurate fall detection system requires rigorous validation and continuous improvement. Inaccurate statements, invalidated results, or failure to acknowledge errors can lead to unsafe applications or reduced trust in our work. Therefore, we will adhere to the principles of honesty, transparency, and continuous improvement throughout the program. We will actively seek, accept, and provide constructive feedback on our technical work, acknowledge and correct errors, and ensure that our claims and estimates are based on accurate and reliable data. In addition, we will give all contributors the credit they deserve to maintain fairness and academic integrity.

5.4.4 Concern about Testing Safety

Prototype testing, especially in projects involving fall detection, carries a risk of injury if not done properly. Participant involvement without adequate precautions can lead to accidents or ethical issues with respect to informed consent. Therefore, to minimize risk, we use ourselves as primary test subjects because we have the expertise to manage potential dangers. We do not involve people outside the team unless appropriate safety measures and risk assessments exist.

References

[1] J. Terven and D. Cordova-Esparza, "A comprehensive review of yolo architectures in computer vision: From yolov1 to yolov8 and yolo-nas," *Machine Learning and Knowledge Extraction*, vol. 5, no. 4, pp. 1680–1716, 2023. DOI: 10.3390/make5040083.

[2] M. Bramberger, A. Maier, B. Strobl, and B. Rinner, "Real-time video analysis on an embedded smart camera for traffic surveillance," in *Proceedings of the 10th IEEE Real-Time and Embedded Technology and Applications Symposium, 2004.*, IEEE, 2004, pp. 174–181.

[3] T. Kasper, M. Ziba, M. Jasiński, P. Kowalski, K. Wojciechowski, and A. Niewiadomski, "A Fall Detection System Using 3D Depth Camera," in *Proceedings of the International Conference on Computer Vision and Graphics (ICCVG)*, Springer, 2012, pp. 762–769.

[4] M. Sokolova and G. Lapalme, "A systematic analysis of performance measures for classification tasks," *Information Processing & Management*, vol. 45, no. 4, pp. 427–437, 2009.

[5] IEEE. "IEEE Code of Ethics." (2016), [Online]. Available: https://www.ieee.org/about/corporate/governance/p7-8.html (visited on 03/11/2025).