ECE 445

SENIOR DESIGN LABORATORY

FINAL REPORT

ECE 445 Final Report

Team #30 Search and Identify

SHITIAN YANG
(shitian.20@intl.zju.edu.cn)
YITAO CAI
(yitao.20@intl.zju.edu.cn)
RUIDI ZHOU
(ruidi.20@intl.zju.edu.cn)
YILAI LIANG
(yilai.20@intl.zju.edu.cn)

Supervisor: Prof. Howard Yang & Gaoang Wang
TA: Enxin Song

May 31, 2024

Contents

1 Introduction

1.1 Purpose

In contemporary household settings, cleaning and organizing often demands substantial time and effort, making the task of remembering and locating items casually placed in a cluttered indoor environment a common daily challenge. For elderly individuals or those with visual impairments and trouble with their legs, this task becomes even more challenging, time-consuming, and frustrating.

In recent years, companies such as OpenAI or Tesla have begun researching home robots like Figure 01[1] and Optimus[2] with large language model. At the same time, models for this type of robot configuration are also in the development stage. Accordingly, we want to study a visual model that supports voice interaction and does not rely on a large language model to provide the robot's item recognition and item positioning functions.

Regarding the current state of most intelligent robots, they predominantly use language-based interaction as a communication aid, but lack means of interaction between humans, robots, and objects. Simultaneously, we are committed to providing visual assistance for robots and robotic arms. This is accomplished in several ways such as voice input, target object recognition, target object distance measurement, and target object contour extraction. Our aim is to enhance the robot's interactive capabilities with real-life events, to assist people of different groups more effectively.

Although there are some existing solutions for locating items, such as using smartwatches or making calls to locate smartphones, or employing tracking devices like AirTags for item retrieval, the existing methods have their own set of limitations.

With the advancement of technology, numerous artificial intelligence systems, including ChatGPT[3], and multimodal models[4]–[6], have been utilized in managing daily and work-related tasks. Inspired by this, the approach we took integrates image capturing and artificial intelligence, aiming to extend the application of AI in various daily challenges. By utilizing these advanced technologies, we can build more convenient human-computer interaction solutions, facilitating a seamless integration of artificial intelligence into daily life, and opening up new possibilities for smart home solutions that are intuitive and effective: A camera based on artificial intelligence large model processing can help people in need to find target objects more intuitively and conveniently in daily life, thus greatly improving the safety and convenience of finding objects in daily life.

1.2 Functionality

The final product of our project is a voice-activated home-use robot designed for item-seeking and navigating. It operates in response to user-initiated voice commands, pro-

viding descriptive information about the desired item for retrieval. The physical design of the system diagram is shown in Figure 1.
In this academic project, we have integrated a variety of frontier artificial intelligence technologies to enhance the robot's environmental recognition and interactive capabilities:

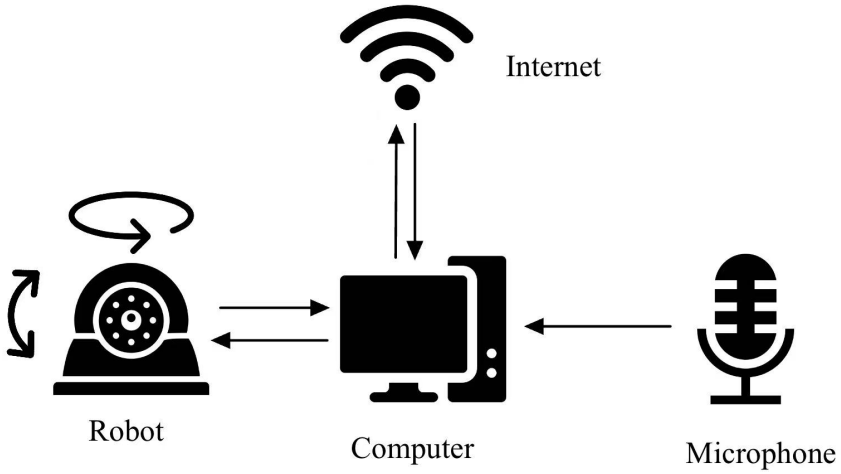

Figure 1: Physical design of the system diagram.

- **Speech Recognition**: Convert the speech to text. We employed the "Whisper Model"[7], a highly efficient deep learning model developed by OpenAI, designed to process and understand speech data in multiple languages.
- **Natural Language Processing**: Our project can understand the users' commands in a sentence and convert them into executable commands. We provided two ways to implement this. The first way is to use the ChatGPT model[3] for the language processing task, which is utilized to refine preliminary prompts given by users into strictly formatted standard prompts suitable for model recognition. Also, we provided local functions to process the record language without the internet, which is free but has restrictions on the structure compared with ChatGPT.
- **Object Detection**: Our project can understand the users' commands in a sentence and convert them into executable commands. We employed the "YOLO" (You Only Look Once) model[8], more specifically its more recent version "YOLO-World"[9]. "YOLO" is a popular object detection and image segmentation model. Due to its fast real-time recognition capability in identifying and locating multiple objects within images, it is widely used for dynamic object detecting, identifying, and tracing tasks. Due to the one-word restriction of YOLO, we employed Clip [10] to enable the detail description function. We utilized Clip to find the most appropriate image(s) with the record sentence from the set of images processed by YOLO.

- **Detail Outline Mask Generator**: By using Segment-anything [11] for the output from Clip, we can get the detail outline mask for the target object.
- **One-click Automatic Scanning**: With Python-integrated control programming, users can control the whole project to finish scanning the environment with just one click.
- **Distance Measurement**: Our project can detect the distance between the target and the camera to help users know the location of the target more precisely.
- **One-Step Rotation**: The users can adjust the mode from automatic scanning mode to the second mode which can rotate to specific angles. After the user inputs the specific angles, the project can be rotated to the corresponding angles.
- **GUI Interaction Interface**: We have developed a user-friendly GUI using Gradio[12], which serves as the primary interaction interface for our project. This interface allows users to interact with our system in real time, providing immediate feedback and adjustments to the search queries based on the visual outputs displayed. It is designed to be accessible from various devices, including smartphones and tablets, enhancing its usability for individuals with mobility impairments.

1.3 Subsystem Overview

1.3.1 Software Components

- **User Voice Recognition System**: Utilizing the "Whisper" model, this module is key in capturing and accurately parsing user voice input. It ensures that the system can understand multiple languages and accents, allowing the robot to serve a diverse user base. This module is the initial point of user interaction and its accuracy directly affects the efficiency and correctness of subsequent modules, forming the foundation for the system's usability.
- **Prompt Processing System**: We provided Internet way and local way to implement language Processing and generate prompts for the Clip model. Employing the ChatGPT model by using api, this module sends the user's record to the internet and ask ChatGPT to transform user's natural language instructions into clear, precise query prompts. The second way is we utilized the Spacy module to deconstruct general grammar and try to make this procedure work for common question structure. These steps are crucial for ensuring that Clip can accurately understand user requirements and identify and classify multiple objects.
- **Object Detection and Selection System**: Integrating the "YOLO" and "Clip" models, this module is responsible for precisely identifying and confirming the specific item requested by the user. These preliminary filtering steps not only enhance the efficiency of subsequent processing but also reduce the potential for misidentification. Usually, YOLO and Clip use 40ms to process each image. The YOLO will first process images from the camera. Sequentially, Clip will select the appropriate images based on the command of the user. It is vital for maintaining system response speed while ensuring operational accuracy.

- **Mask Generation System**: Using the "Segment-anything" model, this module's primary function is to precisely identify objects related to bounding box or target points. Through semantic analysis and detailed image segmentation, this module not only identifies the most matching object but also accurately determines and depicts its specific outline and shape. It is the final link in precisely aligning user instructions with the robot's actual actions, which is crucial for enhancing user satisfaction and the success rate of the robot's operations to get the object.
- **GUI Interaction System** The GUI of our project, designed using Gradio[12], integrates all software components into a user-friendly operational interface, enabling straightforward monitoring and management of the robotic system. Inputs and outputs are converted into visual formats that are easy to interpret, and each major software module is integrated into an interactive GUI for real-time viewing and management. It also allows users to control the robot remotely via public URLs, ensuring accessibility from anywhere within network coverage, and enhancing its mobile compatibility. Therefore, this setup not only simplifies the user's operation with complex functions but also supports real-time feedback and dynamic control, greatly improving the robot's usability and accessibility for users with mobility impairments or the elderly.

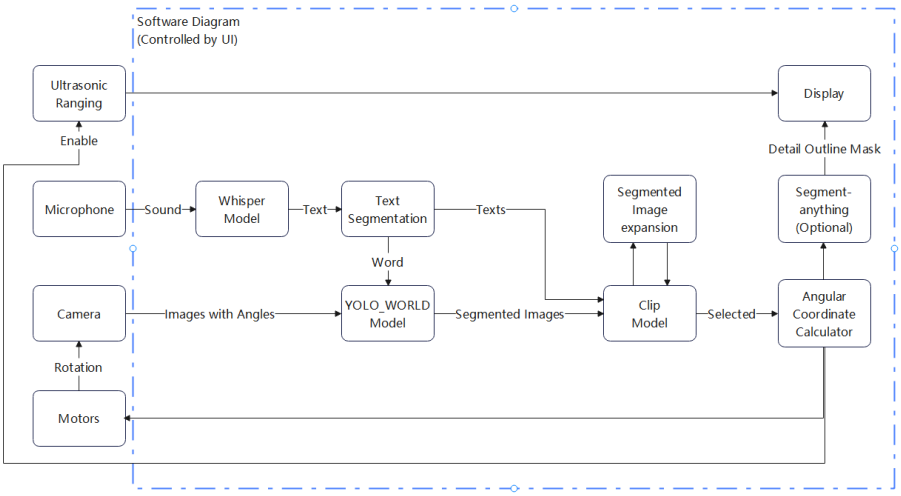

Figure 2: Top level block diagram of the software components.

1.3.2 Hardware Components

- **Drivetrain and Power System**: 360-degree scanning and the ability to rotate to a specific angle in space is achieved by means of twin motors.
- **Control System**: The core component of the hardware part, controls the whole hardware part, including the behavior of the motors and the distance measurement. There are two modes of the motors: mode=0 means when the users input the angles, the motors will rotate to the specific angles and stop at that location; mode=1 means when the users input the angles and the time, the motors will rotate automatically

and scan the surroundings. For distance measurement, there are also two modes: users can choose to measure the distance once or continuously.

- **Gadgets system**: It includes the sonic distance measurement component which can measure the distance according to the commands from the control system.
- **Port Simulation System**: We write a Python program to replace the serial port assistant so that we can integrate all the commands. The data can be achieved for the software to analyze. For the users, they can control the project with one click.

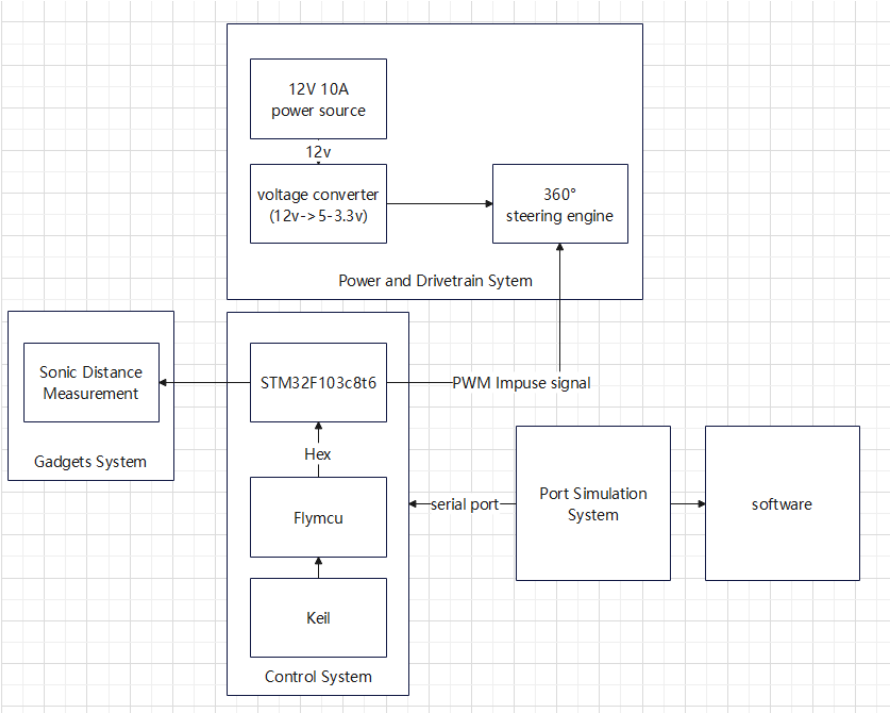

Figure 3: Top level block diagram of the hardware components.

2 Design

2.1 Design Alternatives

2.1.1 Software Components:

- **Microphone Voice-to-Text Conversion System**:
  As will be elaborated more in the verification section, our method tried out different Whisper models[7] together SPHINX[13] to realize the function of voice-to-text conversion. Namely, the four models are Whisper tiny, Whisper base, Whisper small, and Whisper medium. While they have an increasing order of accuracy, their run time increase as well. So to balance performance, we used the Whisper small model. Compared with other voice-to-text conversion models, the Whisper model provides an outstanding balance of accuracy and efficiency. It does not require specific fine-tuning and can function at scale. Most of all, it has pleasing accuracy, especially in our circumstance of basic everyday communication.

- **Object Detection System**:
  For the function of object detection, we also considered various methods such as Region-based Convolutional Neural Networks (R-CNN) [14] and Glip [15]. R-CNN is not as fast as YOLO-World for real-time applications and require substantial computing resources to run. For Glip, though its performance is stronger than Clip and it is a new model, it is hard to set up. We found that most of the modules required by Glip were out of date. It needs a long time to find the appropriate version to make them workable for Glip.

- **Object Mask Generation System**:
  For the function of object mask generation, we initially thought of merely adopting AbsVit and Segment-Anything. However, after testing with multiple objects of similar conditions, we found that AbsVit was sensitive to the module version, hard to maintain, low definition, low generating speed, and has a clash with other parts requirements. We then proposed to use Segment-Anything only.

- **Calibration and Computation System**:
  This system is mainly based on geometric calculations, and few alternatives can be done on the math. However, there was indeed consideration on the choice of the camera, we could use cameras of different AFOV, such as 40 degrees, 60 degrees, or 70 degrees, and we could also choose cameras of different focal lengths. We eventually chose the camera with an AFOV of 60 degrees and an adaptable focal length. The reason is that while a smaller AFOV can be more accurate in calculating the specific angle of the desired object, it will require more rotations of the motor and more time. To ensure efficiency and still retrieve a relatively high accuracy, the AFOV of 60 degrees was chosen.

2.1.2 Hardware Components:

- **Drivetrain and Power System**:
  Alternative Approaches:
  1. 360° rotation bulbous bossing.
  2. Two-axis motors rotation.
  Chosen final approach:
  Two-axis motors rotation.
  Reason:
  Two-axis can achieve more attitude control, For instance, we can control one motor to be still and make the other motor to rotate to provide a smoother spinning solution. Its control can be achieved through Python (through serial port control) so that integrated control can be finished.

- **Gadgets system**:
  Alternative Approaches:
  1. Calculate the distance through photos by LLM.
  2. Sonic ranging system.
  Chosen final approach:
  acoustic ranging system.
  Reason:
  Physical distance measurement is more accurate and faster, but LLM takes more time and relies on the accuracy of photo-shoots, making it difficult to meet daily life's needs. Sonic distance measurements cost less than LLM as LLM requires higher requirements for computer hardware components.

- **Control System**:
  Alternative Approaches:
  1. Through STM32F103C8T6 microcontroller to control the whole hardware component.
  2. Through FPGA to control the whole hardware component.
  Chosen final approach:
  Through STM32F103C8T6 microcontroller to control the whole hardware component.
  Reason:
  Before commands are inputted through serial ports, we should use the software to burn the program into the control panel. Keil and Flymcu for STM32F108C8T6 microcontroller are convenient choices for controlling. Besides, serial port control can be realized through Python thus enabling integrated design. Meanwhile, microcontroller control can provide a low-cost project design.

- **Port Simulation System**:
  Alternative Approaches:

1. Multi-command line control for serial ports.
2. Programming in Python to control the serial ports and input commands.
3. Input a single command through serial ports.

Chosen final approach:
Programming in Python to control the serial ports and input commands.
Reason:
Presetting all the required commands in advance to automate the process with a single keystroke can give the users a much simpler control method. Data can be quickly and easily imported into pre-trained models for analysis.

2.2 Design Description & Justification

2.2.1 Microphone Voice-to-Text Conversion System:

For our project, we will utilize the Whisper model [7] and the SPHINX [13] to implement the Microphone Voice-to-Text Conversion feature. Whisper is a groundbreaking speech recognition model that has been trained on a vast corpus of audio transcripts from the internet, amounting to 680, 000 hours of multilingual and multitask supervised learning. This extensive training enables the Whisper model to deliver high-quality speech recognition capabilities in a zero-shot transfer setting, effectively eliminating the need for dataset-specific fine-tuning. Remarkably, Whisper approaches the accuracy and robustness of human listeners and is designed to handle a wide array of speech-processing tasks. These include multilingual speech recognition, speech translation, and spoken language identification, facilitated by its transformer sequence-to-sequence model architecture.

2.2.2 Language Processing and Prompt Generation:

The way to connect the Internet: We used API to connect the ChatGPT and asked ChatGPT 3.5 to help us split and restructure the command from users to generate prompts.
The way only use local environment: we decided to utilize a Python module called Spacy which is a language module to identify word classes like nouns, adjectives and prepositions. Then we try to create some regularization structures to fit the general question structure for users. After splitting the structure, we will generate prompts by changing each part of the structure to add a noise prompt for Clip.
The regularization structure is like this: (*prefix*)+(*adjective*)+(target noun)+(*direction preposition*)+(*adjective*)+(reference noun)+(*suffix*), where "*" represents multiple words. Hence, there are four parts for the prompts, which enable Clip to identify the detailed environment step by step. The four parts are: the adjective part before the target noun(object), the adjective part for the reference noun(object), the direction preposition, and the reference noun.

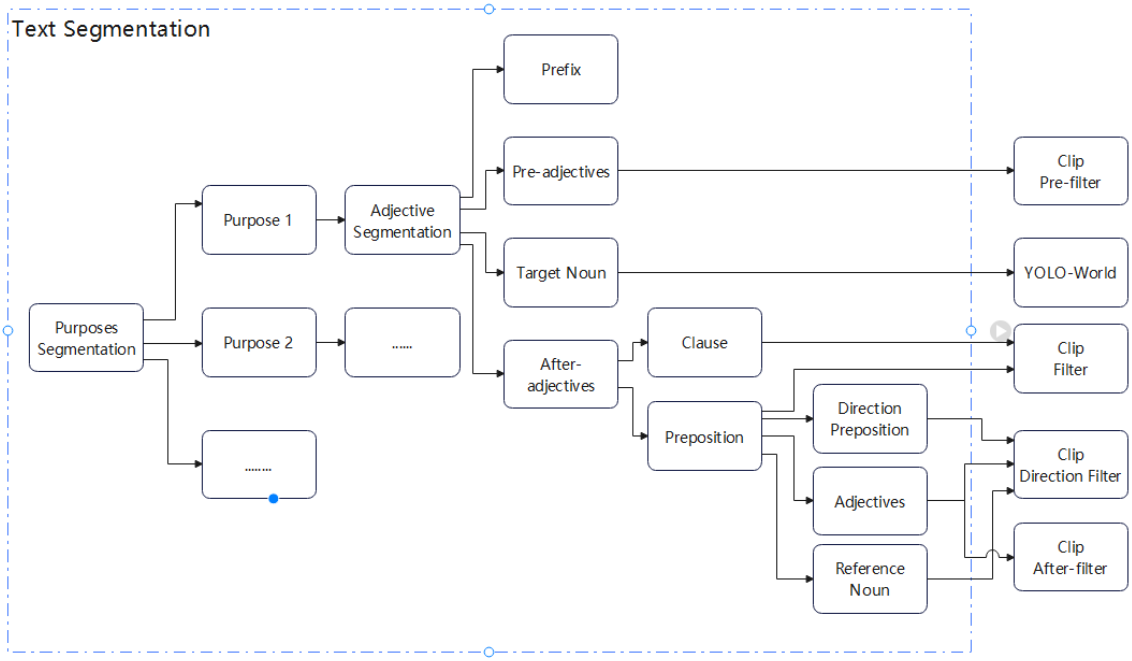

Figure 4: Diagram for language processing

2.2.3 Object Detection System:

The Object Detection Module is tasked with initiating preliminary searches to identify potential objects of interest within captured environmental images. This module employs the YOLO-World and GLIP vision models to conduct object detection tasks across a range of lighting conditions. Its function is to rapidly sift through segmented regions of the environmental images, isolating areas that may correspond to the target objects, thereby facilitating the generation of more precise object masks in subsequent stages.

First, we input the extracted target noun into YOLO-World. YOLO-World then processes images from all perspectives, segments all objects, and labels each with a bounding box and the name of the corresponding image file. Subsequently, we provide prompts generated from the adjective part preceding the target noun to Clip for further screening of the images processed by YOLO-World. Here, the pass-through rate is adaptively adjusted based on the level of detail specified in the user's instructions, ensuring that overly stringent restrictions do not exclude potential candidates. Following this, we expand the segmented images to broaden the field of view, enabling Clip to detect information related to the reference. This expansion is carefully controlled; if the referenced noun and the target noun are not the same, we avoid including additional candidates in the expanded images, which could present multiple options to Clip. Additionally, the direction of expansion is adjusted based on directional prepositions. Finally, we sequentially provide Clip with prompts for the adjective part associated with the referenced noun, the directional prepositions, and the referenced noun itself, ultimately filtering to identify the most likely image or images.

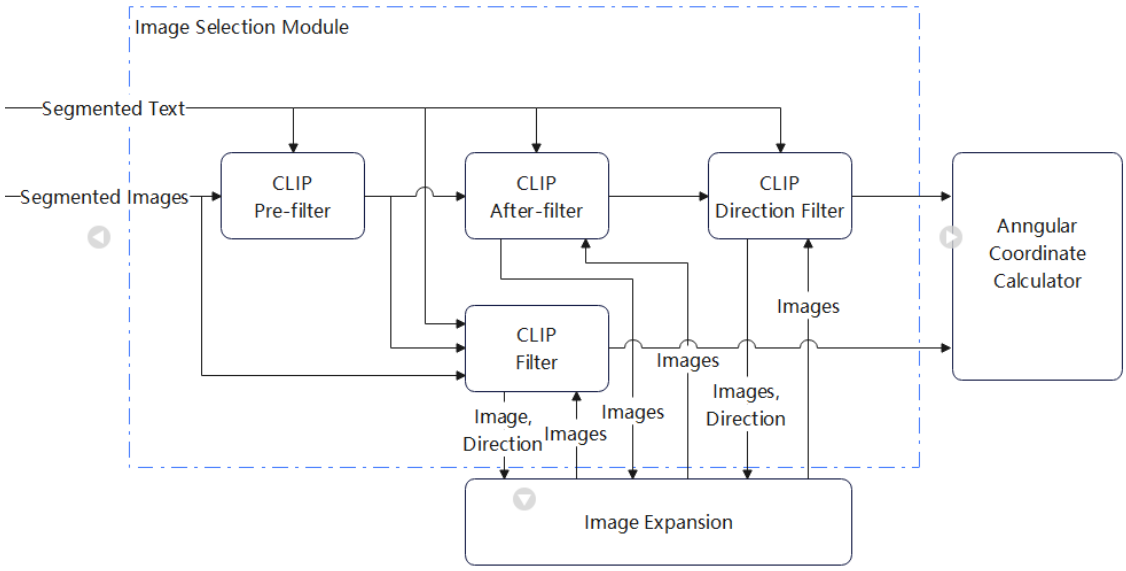

Figure 5: Diagram for image selection by using CLIP.

2.2.4 Object Mask Generation System:

The Object Mask Generation (OMG) System is designed to produce high-quality masks that enable the precise extraction of target objects from their backgrounds, intended for further verification and display purposes. The system accepts a segmented image accompanied by a bounding box and a textual description of the desired object and generates masks for the target object based on these input parameters.

In order to generate high-quality masks for the target object, we utilize the Segment-Anything model [11] (SAM) developed by MetaAI for this purpose. SAM adopts a universal segmentation approach that recognizes contextual information and detects object boundaries, enabling it to effectively identify and isolate specified objects from their surrounding environments without having to include them in the training dataset.

However, SAM has limitations in classifying and understanding the objects especially when a detailed description is provided. Hence, we decided to use it for the final images outputted by the object detection system. Also, usually, SAM needs 1.5 minutes to process each image and can work individually. As a result, we will make it as an optional choice after the whole procedure.

As explained above, the object detection system will pass the image, text prompt, and the bounding box of the object to the object mask generation system. The object mask generation system will output a clipping mask of the clipped object. We then use it to remove

the background and extract the object for display and verification.

2.2.5 Calibration and Computation System:

In order to navigate the user to their desired object, our system must analyze the direction of the target, which can be indicated with coordinates of a polar angle and an azimuth angle. This module receives coordinates of the camera direction when capturing the image, along with the bounding box of the target object. Utilizing this data, the system computes the directional coordinate of the target object's center point relative to the camera. Subsequently, it integrates these calculations with the camera's direction to derive the object's absolute directional coordinates.

The system employs geometric calculations to transform the object's relative position in the image into real-world directional coordinates. The input data includes the camera angular field of view (AFOV) and its directional coordinates (ϕ, θ), the WIDTH and HEIGHT of the image in pixels, and the bounding box of the target with parameters (x, y, w, h).

We apply the following calculations:

1. Target Center Coordinate and Focus Length:

$$x_c = x + \frac{1}{2}w \qquad y_c = y + \frac{1}{2}h \qquad F = \frac{\text{WIDTH}/2}{\sin(\text{AFOV}/2)}$$

2. Relative Target Direction:

$$\varphi' = \sin\left(\frac{x_c - \text{WIDTH}/2}{F}\right) \qquad \theta' = \sin\left(\frac{\text{HEIGHT}/2 - y_c}{F}\right)$$

3. Target Direction:

$$\Phi' = (\varphi + \varphi') \mod 2\pi \qquad \Theta' = \theta - \theta'$$

Here the AFOV is the camera's angular field of view on its x axis, the coordinate system of the image has the x-axis toward the right and the y-axis downward. The spherical system adopts that polar angle θ is measured between the radical line of the target center and the upward direction, and the azimuthal angle φ is measured between the orthogonal projection of the radial line of the target center onto the horizontal plane and the x-axis.

The geometric principle of the above calculation is shown in Figure 6.

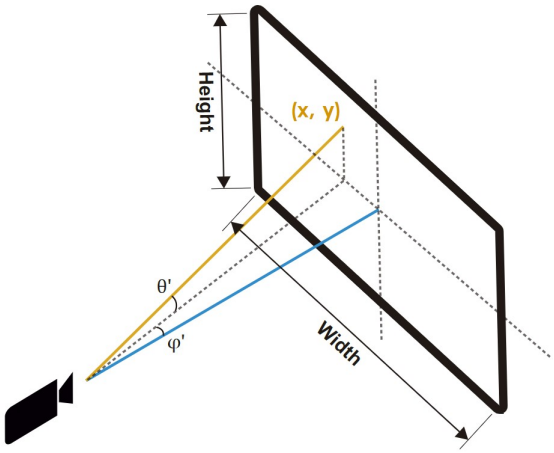

Figure 6: Geometrical explanation of the directional coordinate calculations

2.2.6 GUI Interaction System

The GUI Interaction System is engineered using Gradio[12], a robust framework for building interactive, web-based interfaces for machine learning models. This system translates the complex software processes of our robotic system into a user-friendly, accessible web interface, facilitating seamless user interaction through technical integration.

- **Implementation Framework**: Developed with Gradio, the GUI operates as a front-end application that communicates with our backend systems via HTTP requests. The GUI is structured into modular components corresponding to each major software function, such as voice recognition, object detection, and mask generation. This modular design allows for easy updates and maintenance without disrupting the entire system.
- **Data Handling, Processing, and System Integration**: User inputs, such as voice commands or selection actions, are captured through interactive elements in the GUI and sent to the backend where they are processed. The results, including detected objects and generated masks, are dynamically displayed in the GUI. These interfaces facilitate robust data exchange between the Gradio interface and the core software components, such as the Whisper model for voice recognition, the YOLO model and CLIP model for object detection and SAM to generate mask. This integration ensures a smooth and responsive data flow, optimizing system performance and user experience.

2.2.7 Drivetrain and Power System:

The drivetrain and power system mainly consist of two parts, a power source with a voltage converter and a two-motor system. The 12v power supply provides a stable 12V voltage and 10A current, and inputs into the voltage converter, which will then be converted to 5V and 3.3V with ground voltages respectively. A specific requirement for the voltage converter is that it must supply stable current and voltage within a limiting re-

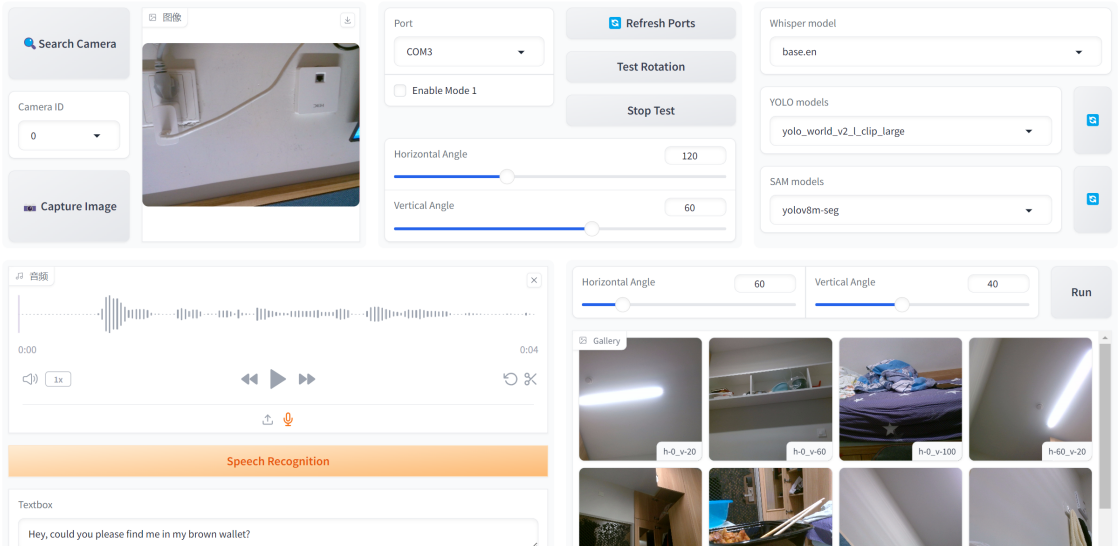

Figure 7: Graphical User Interface Effect

gion of ± 0.1V to ensure the normal operation of the system.

We designed and soldiered the PCB of the voltage converter ourselves with a schematic shown in the following Figure 8. From actual testing measurements with a multimeter, with an input of 12V and 10A the output voltage was 4.988V for the 5V port and 3.320V for the respectively.

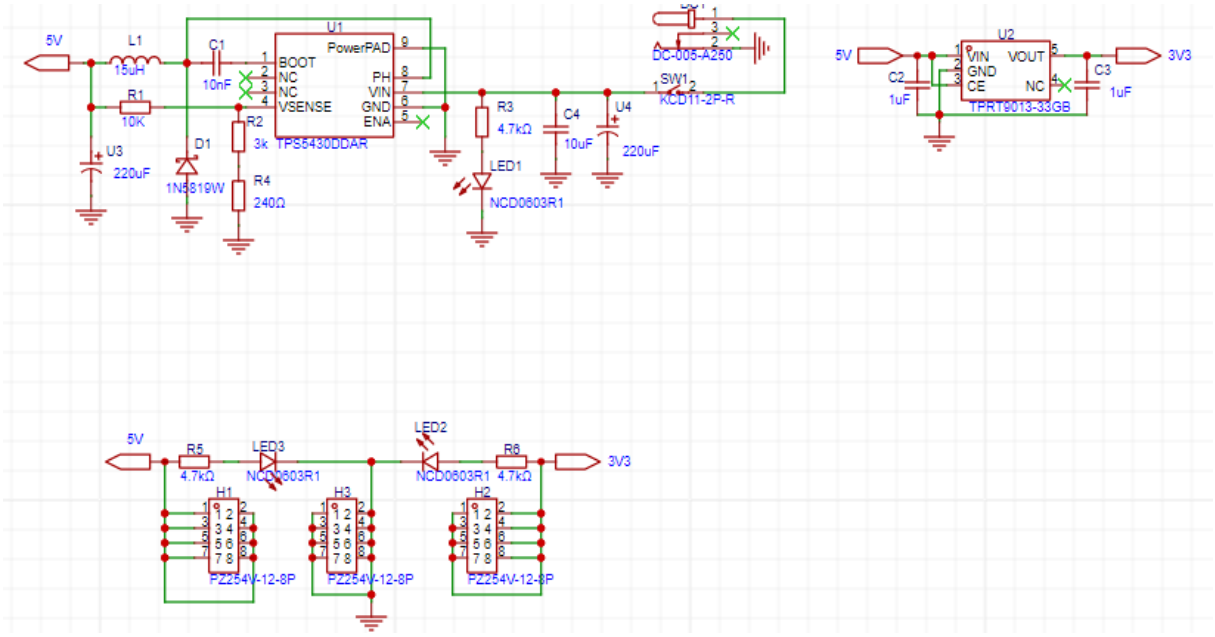

Figure 8: Schematic of our originally designed voltage converter

For the two-motor system we designed, one is used for horizontal rotations while the

other is used for vertical rotations. An STM32 microcontroller is used for processing the command being inputted and then controlling the motors to rotate desired angles. More descriptions of STM32 will be elaborated in the next section of the "STM32 Control System". In controlling the motor, we manage to set the angle by modifying the duty cycle of PWM. 0 to 360 degrees of rotation corresponds to pulses from 500 μs to 2500 μs, so the pulse needed can be calculated using the following Equation 1.

$$Pulse = 500 + angle \times 5.55555555 \tag{1}$$

The motor can accept 5V voltage and can carry up to 30kg items. A wireless camera and a sonic detector are also attached to the top of the motor for real-time data capturing. Figure 9 is the structure design of the two-motors system.

2.2.8 STM32 Control System:

This system mainly conducts the following work: Hardware code is written and compiled using the software of Keil5, and then the compiled code is downloaded and written onto the STM32 board using FlyMcu. Through hardware programming, we ensure the motor can rotate correct angles according to the set target. After rotating a specific angle, it stays for a predetermined amount of time, and then automatically repeats rotation, until a full cycle is completed.

2.2.9 Gadgets System:

The gadget system consists of two components: A flashlight indicating the direction of the desired object in the horizontal plane and a sonic distance calculation module.
For the sonic module, a schematic revealing connection with the STM32 board is shown below in Figure 10. The Trig port will generate pulses greater than 10us, while the Echo port will capture a signal that bounces back from the object. The time interval between the high-level signals will be calculated as the traveling time of the sonic wave. Distance is then calculated using the following Equation 2.

$$Distance = time_{diff} \times 340/2 \tag{2}$$

The sonic sensor is attached closely to the camera, and the two can rotate together to detect objects. Only when the camera finishes scanning and has found a specific angle aiming at the desired object will the sonic sensor be turned on to measure the distance.

2.2.10 Port Simulation System:

This module is used in connecting with the software components and informing the camera about when to capture photos. Python modules of pyserial and cv2 are adopted. With a corresponding port and baud rate set, the messages to be inputted to the STM32 will be encoded in the form of ASCII code. The decoding process also adopts the form of ASCII code, and all data read will be returned in the form of a thread. Whenever the

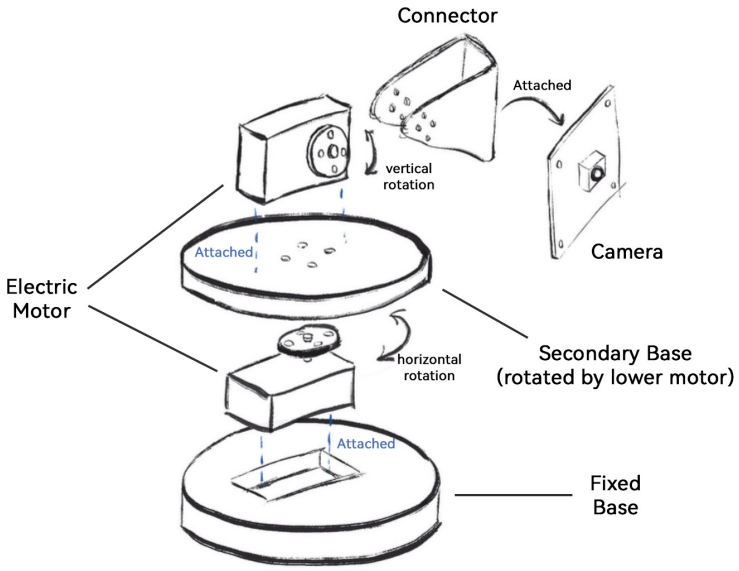

Figure 9: Mechanical structure of our robot's two-motors system

data read from the thread senses a change in the angle of the vertical motor, which means that the motor has rotated a particular angle, the camera will capture photos with the corresponding motor angles labeled. When photos from all set angles are captured and stored properly, the software components will process and by using the calibration and calculation system, a specific angle directed to the object will be returned. Such data will then be used to rotate the motors and finally returning the direction.

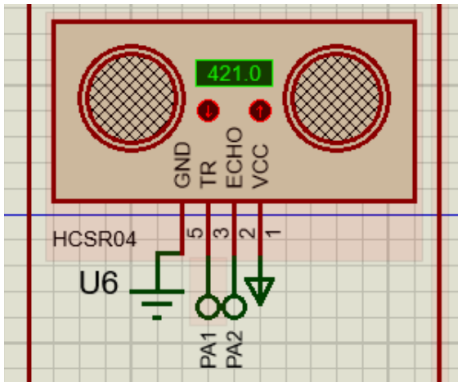

Figure 10: Port connection of the sonic sensor

3 Cost & Schedule

3.1 Cost Analysis

Our work is to be estimated 10 hours/week for 4 people. One person is about $ 30/hour, we plan to finish ECE 445 design this semester for 16 weeks:

$$\frac{\$30}{hour} \times 4 \times \frac{10hours}{week} \times 16\,\text{weeks} \times 2.5 = \$48000$$

Table 1: Cost analysis

Part	Mft	Desc	For	Price	Qty	Total
Steering Engine	DS	360°, 3.3 V	Camera	188	2	376
STM32F103c8t6	DS	3.3 V	Steering Engine	28	1	28
Power Source	XMS	12 V	Steering Engine	14.5	1	14.5
Converter	SD	12 V $\rightarrow$ 3.3 V	Steering Engine	10	1	10
UVC Camera	HIKVISION	3.3 V	Software	300	1	300
Laser Pointer	HD	12 V	Gadgets	58	1	58
Sonic Sensor	DS	12 V	Gadgets	32	1	32
Total	/	/	/	/	/	818.5

3.2 Schedule

- **Mar 25- Mar 31**
  Shitian Yang:
  Download, install, and locally deploy the Whisper model. Installed AbsVit and Segment-anything models and understand their API calls.

  Yitao Cai:
  Prepare the environment for Segment-Anything model, GLIP model, YOLO model. Investigate UVC camera control protocol with OpenCV.

  Ruidi Zhou:
  Mount the camera onto our motor and achieve basic code-controlled rotation. Basic hardware testing.

  Yilai Liang:
  Completed soldiering and wiring on the PCB board, help Ruidi conduct the hardware testing. Start designing the voltage converter.

- **Apr 1 - Apr 7**
  Shitian Yang:
  Perform API calls with the Whisper, AbsVit, and Segment-anything models to ensure they can correctly execute tasks according to our requirements. Begin attempts to streamline the process.

  Yitao Cai:
  Test the functionality of the selected object detection models with images from the dataset and real-world collected images. Implement the camera control program and test it.

  Ruidi Zhou:
  Purchase, mount, and code-control the second motor. Achieve the 360-degree rotation ability with testing.

  Yilai Liang:
  Read papers about 3D image reconstruction. Set up the environment and complete testing on a single image.

- **Apr 8 - Apr 14**
  Shitian Yang & *Yitao Cai*:
  Integrate the entire workflow involving the Whisper and object detection models selected from the models mentioned above based on their quality, and determine the appropriate parameters for each.

  Ruidi Zhou & *Yilai Liang*::
  Design and implement the stretchable base of our robot. Explore the possibility of our dynamic system.Explore the data transmission with the camera, perform object reconstruction with real objects using all methods, and choose the most suitable.

- **Apr 15 - Apr 21**
  Shitian Yang & *Yitao Cai*:
  Continue the tasks from the previous week to establish the software workflow, test its functionality and performance, and modify it if required.

  Ruidi Zhou& *Yilai Liang*:
  Implement the hardware code of distance calculating, and complete hardware connections.

- **Apr 22 - Apr 28**
  All four members meet together to connect our software and hardware components together. The construction of the mechanical structure of the two motors with cameras is also done this week. Design a GUI to control the whole procedure.

- **Apr 29 - May 5**
  All four members meet together for the round of testing and sharing information and data about the whole implementation process. Making minor fixes such as tightening the wire connections and adjusting camera's initial angles.

4 Requirements & Verification

4.0.1 Microphone Voice-to-Text Conversion System:

Given that our inputs come through a microphone, we anticipate minimal background noise. In tests conducted on the LibriSpeech.test-clean dataset [16], the Whisper model variants exhibit impressive accuracy: Whisper tiny (approximately 1GB in size and 32x relative speed) shows an error rate of 5.6%, Whisper base (approximately 1GB in size and 16x relative speed) has a 4.2% error rate, and Whisper small (approximately 2GB in size and 6x relative speed) improves further to 3.1%. Given our focus on everyday basic communication, we expect even higher accuracy levels, aligning with our high-level requirements. Moreover, thanks to its extensive training across various languages and datasets, Whisper demonstrates low sensitivity to accents, making it well-suited for a broad user base.

Model	**Test on original dataset**	**Test on recordings**		**Run time**
	Accuracy	**Accuracy**	**False language**	
whisper_tiny	94.4%	94.2%	15%	1.2s
whisper_base	95.8%	95.1%	8%	2s
whisper_s	96.1%	95.2%	8%	3.6s
whisper_m	96.5%	95.1%	6%	5s

Table 2: We tested four models of Whisper. The "False language" refers to the occurrence of non-English words generated by the multi-language model due to our accent or the transformation of short sentences.

4.0.2 Object Detection System:

By integrating two models of YOLO-World as mentioned above, we aim to ensure the precision and speed of our Vision Module, meeting our high-level requirement for accuracy in vision. Moreover, leveraging ChatGPT's linguistic intelligence, we intend to design smart feedback mechanisms for scenarios where objects are not detected or only similar items are found. This intelligent interaction with users will clarify the current recognition status and address any issues encountered, thereby enhancing the user experience by providing insightful and constructive feedback. A testing is shown in Table 3

4.0.3 Object Mask Generation System:

During our test, we found out that SAM may fail to find the desired object if only the text description of the object is provided, especially if the object is unique and has not been

Objects	YOLO-World_m					
	No occlusion		25% occlusion		50% occlusion	
	Recall	False Alarm	Recall	False Alarm	Recall	False Alarm
bottle(10)	1	0.09	0.8	0	0.4	0
book(5)	1	0	0.8	0	0.3	0
scissor(5)	1	0	1	0	0.8	0
pen(10)	1	0	1	0	0.7	0
Objects	YOLO-World_x					
	No occlusion		25% occlusion		50% occlusion	
	Recall	False Alarm	Recall	False Alarm	Recall	False Alarm
bottle(10)	1	0.09	0.7	0	0.2	0
book(5)	1	0	0.8	0	0.3	0
scissor(5)	1	0	0.8	0	0.2	0
pen(10)	1	0	1	0	0.6	0

Table 3: The ability of two YOLO_World models was tested in a messy room in different occlusion situations. We placed 5 or 10 objects for testing. The image of test room you can find below.

trained before or if multiple objects of a similar kind confuse the model. Therefore we decided to separate the task into multiple parts, to utilize other models with strength in object detection and let SAM only handle the task of generating object clipping masks.

4.0.4 Calibration and Computation System:

In order to actualize this function, a prerequisite is the precision of the input parameters obtained by the module. This encompasses the intrinsic parameters of the camera, such as the AFOV and focus length, as well as the coordinate parameters of the camera orientation acquired from the mechanical structure, which must be precisely aligned with the camera's actual physical direction under the control of the mechanical structure. This calibration is crucial for subsequent computations.

From the datasheet of the camera, we obtained the AFOV of the camera to be 60 degrees. The verification process is done on the basis of the drivetrain and power system, and it's also the final performance of our whole project.

Objects	AbsVit		
	No occlusion Accuracy	25% occlusion Accuracy	50% occusion Accuracy
bottle(10)	0.9	0.9	0.6
book(5)	1	1	0.8
scissor(5)	0.8	0.8	0.8
pen(10)	1	1	0.7

Table 4: The ability of the AbsVit model was also tested in a messy room in different occlusion situations. Compared with YOLO, AbsVit uses a sentence as a prompt, so we focus on the accuracy of the target object with detailed description.

4.0.5 Drivetrain and Power System:

For this model, two tests were conducted: one focused on whether the motor rotated desired angles when implementing a certain angle, the other focused on whether the voltage converter we designed had appropriate outputting voltages and currents meeting our expectations. By tightly fixing a spinner to the top of the motor, the angle of the motor can simply be indicated and measured. By using a protractor, we measured the rotation of the motor to fit our implemented angle perfectly with a fault tolerance within 0.1 degrees with all three attempts.

The requirement for the voltage converter is that it can provide stable current and voltage (3A, 5V and 1A, 3.3V respectively) when being supplied with 12V, 10A from the power source. The tolerance should be $\pm$0.1V for voltage while $\pm$0.05A for the current to be considered successful. We adopted a multimeter for measurement as shown in Table 6.

Output port	**Measured Voltage**	**Measured Current**
port of 5V	4.992V	3.00A
port of 3.3V	3.320V	0.99A

Table 5: Voltage and Current verification.

From the result, it is proved that our implementation was successful.

4.0.6 Gadgets System:

The requirement and verification of this system mainly center around the sonic sensor. Due to noise and distraction in the environment and the fact that targeted objects may

have their unique shapes unconsidered in advance, we set the fault tolerance to be within 5% of the actual distance. Our attempts focus on comparing the measured result from the sonic sensor to the result of a distance meter when the sonic sensor is mounted to the motor and near the camera, the difficulty with this is that it is difficult to measure the exact starting point of the sonic sensor so the error may occur at around 1mm. Also, different points from the sensor may result in different results especially when encountering a round surface. Details of the testing are shown in Table

Equipment used	**Attemp 1**	**Attemp 2**	**Attemp 3**
Distance Meter	0.089m	0.972m	3.746m
Sonic Sensor	0.092m	0.985m	3.781m
relative difference	3.26%	0.71%	0.93%

Table 6: Comparison of the distance meter and sonic sensor under different distances.

From the comparison, it's obvious that as distance increases, the absolute value of the resulting measurement difference increases by a little, however, the relative difference or relative error becomes very negligible and less than 1% when the distance to the object is large.

5 Conclusion

5.1 Accomplishments

Functional Achievements
Our project has successfully implemented several key functionalities that enhance the user experience and operational efficiency of our home-use robot. These include:

Custom Angle Rotation of Servos: We have achieved precise control over two servos, allowing for custom rotational angles that are essential for accurate camera positioning and object targeting.
Camera Module Operation: The camera module effectively captures photographs from various angles, storing them for subsequent processing. **Target Object Distance Measurement**: The system accurately
measures the distance to the target object, which is crucial for navigating and retrieving the item.
Voice-to-Text Conversion: Utilizing advanced speech recognition technology, the system converts spoken commands into text, facilitating seamless human-robot interaction.
Candidate Identification for Item Retrieval: The robot identifies potential candidates for the target item based on initial visual data.
Image Filtering Based on Detailed Descriptions: Using detailed descriptions such as shape, contour, color, and relative position, the system filters and selects the most relevant images.
Detailed Outline Mask Generation: For the final selected image, the system generates a detailed outline mask, which is critical for precise object extraction and display.
Real-World Angular Coordinate Calculation: The system calculates the real-world angular coordinates of the target object based on the final image, enabling accurate directional guidance.

Overall System Integration and GUI Interaction
From a holistic perspective, the project integrates these functionalities into a cohesive system controlled via an intuitive Graphical User Interface (GUI). The process begins with the user vocalizing the need to locate an item. Following this:
Servo and Camera Operation: The servos rotate to position the camera, which captures and stores images from different directions.
Image Processing and Segmentation: YOLO-World processes these images to segment and identify the target object. Detailed Filtering with Clip: Based on the user's detailed descriptions of the target item, Clip refines the selection by filtering out irrelevant images.
Directional Guidance and Final Object Retrieval: The system calculates the real-world angular coordinates from the final selected image. It then controls the servos to orient the indicator towards the location of the target item.
Optional Detailed Contour Functionality: Users have the option to activate the detailed outline mask generation feature for enhanced accuracy.
GUI Parameter Adjustments: Users can adjust parameters within the GUI to optimize the system's performance based on specific scenarios.

5.2 Uncertainties

5.2.1 Software Limitations

Image Model Constraints: YOLO-World: This model struggles with unrecognized vocabulary (e.g., ”iPad”) and has limited ability to identify objects that are obscured or partially hidden. It is also sensitive to specific keywords, where similar objects might not be recognized correctly (e.g., confusing ”kettle” with ”bottle”). Clip: The performance of this model can be affected by environmental factors such as lighting and shadows, which may alter the perceived color and shape of objects.
Voice-to-Text Model Limitations: Whisper: While it supports multilingual translation, its accuracy diminishes in noisy environments or with heavily accented speech, which can lead to misinterpretations or errors in command execution.
Semantic Segmentation Limitations: Without integration with ChatGPT, the system may struggle with complex sentence structures that deviate from the expected format: (prefix)+(adjective)+(target noun)+(direction preposition)+(adjective)+(reference noun)+(suffix). Idiomatic expressions, complex clauses, and unusual locational phrases may not be processed effectively.
Process Flow Errors: During the operational process, discrepancies in weight and recognition accuracy might lead to the exclusion of potential candidates. For instance, in a scenario with multiple red bottles under varying lighting conditions, the system might fail to identify all relevant objects due to significant differences in recognition confidence levels.

5.2.2 Hardware Limitations

Servo Precision: Repeated rotations and inherent mechanical limitations may lead to inaccuracies in achieving the desired angular positions, affecting the overall precision of the system.
Camera Center Shift: Frequent use can loosen the camera mount, causing instability during rotation. This results in non-level photography and a shift in the camera’s central alignment.
Connection Uncertainty: Prolonged use may lead to wear and relaxation of USB ports, wires, and hardware interfaces, potentially causing malfunctions or failures.
Hardware Damage: Continuous operation can lead to overheating and potential damage to the hardware components, necessitating replacements.
Wire Entanglement: Multiple rotations might cause wires to tangle or obstruct the camera’s view, complicating operations and maintenance.
Positional Accuracy: Optical distortions, such as fisheye effects, are not completely eliminated, leading to discrepancies between the pixel representation in images and the actual world coordinates.

5.3 Future Work / Alternatives

1. Our mechanical design can be modified to be more firm and portable. A stronger mechanical structure is designed to increase overall stability, and a wire management device is used to rationalize the arrangement of DuPont wires. In that case, the users can move the project to any position they want.
2. Our project is immovable, so the four-wheel motion device can be added at the bottom, which can automatically move towards the target according to the output angle and distance, and can automatically avoid obstacles in the process, in this way, the robot can go the target position to fetch the target items for the users.
3. Adding a robotic arm device can determine the means of grasping based on the distance and the contour of the object, and recognize the object found based on the voice description. If the costs can also be lowered, our project can be large-scale used in household robotics, item sorting, freight transportation, and manual manufacturing.

5.4 Ethics and Safety

There are indeed several concerns about safety and ethics with our project. First of all, a flashlight is considered to be mounted to the camera for target directing. To address such issues, we will always keep the power off during testing and only turn on the flashlight when demoing and final locating at a relatively low level to comply with relevant safety regulations and to minimize the risk of harm to users or bystanders[17]. While our design also adopts electric power sources and motors, special care will be paid to robust testing and validation procedures to ensure the reliability of the system and to prioritize user safety. This complies with the ACM Code of Ethics, Section 2.9, that "Design and Implement Systems That Are Robustly Secure[17]."

As a project involving visual and vocal data utilization, it's crucial that such data is handled securely and with respect for user privacy. With the scope of the course ECE 445, our team members will mainly be the operators and users, and we will take on the responsibility not to use or spread others' data without formal and proper permission[18].

Another concern arises in transparency and explainability. Even if users permit our usage of their vocal data, it's our unshirkable duty to provide clear explanations of its decision-making processes, especially regarding object recognition and task execution, to ensure users understand and trust the system's behavior[17].

Last but not least, to ensure the users' safety and convenience, we adopt a battery management system that will monitor the charge on all our electric components since they are working wirelessly. This design not only provides the users' information about the charging condition of our robot but can also warn ahead if something unexpected or unsafe is about to occur, which also complies with Section 2.9 of the ACM Code of Ethics [17].

All of our group members carefully affirm that we will strictly follow the IEEE and ACM Code of Ethics.

References

[1] Figure, *Figure status update - openai speech-to-speech reasoning*, Mar. 2024. [Online]. Available: https://www.youtube.com/watch?v=Sq1QZB5baNw.

[2] Tesla, Inc., *Optimus: General-purpose robotic humanoid*. [Online]. Available: https://www.tesla.com/AI.

[3] OpenAI, *Chatgpt can now see, hear, and speak*, OpenAI Blog, Accessed: 2024-01-10, 2024. [Online]. Available: https://openai.com/blog/chatgpt-can-now-see-hear-and-speak.

[4] S. Antol, A. Agrawal, J. Lu, *et al.*, *Vqa: Visual question answering*, In Proceedings of the IEEE International Conference on Computer Vision, 2015.

[5] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, *End-to-end object detection with transformers*, 2020.

[6] B. Shi, T. Darrell, and X. Wang, *Top-down visual attention from analysis by synthesis*, 2023. [Online]. Available: https://arxiv.org/abs/2303.13043.

[7] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, *Robust speech recognition via large-scale weak supervision*, 2022. arXiv: 2212.04356 [eess.AS].

[8] Ultralytics. ""You Only Look Once model by Ultralytics"." (2023), [Online]. Available: https://github.com/ultralytics/ultralytics (visited on 04/08/2024).

[9] T. Cheng, L. Song, Y. Ge, W. Liu, X. Wang, and Y. Shan, *Yolo-world: Real-time open-vocabulary object detection*, 2024.

[10] A. Radford, J. W. Kim, C. Hallacy, *et al.*, *Learning transferable visual models from natural language supervision*, 2021. arXiv: 2103.00020 [cs.CV].

[11] A. Kirillov, E. Mintun, N. Ravi, *et al.*, *Segment anything*, 2023. arXiv: 2304.02643 [cs.CV]. [Online]. Available: https://arxiv.org/abs/2304.02643.

[12] A. Abid, A. Abdalla, A. Abid, D. Khan, A. Alfozan, and J. Zou, "Gradio: Hassle-free sharing and testing of ml models in the wild," *arXiv preprint arXiv:1906.02569*, 2019.

[13] K. F. Lee, H. W. Hon, and R. Reddy, *An overview of the sphinx speech recognition system*, Jan. 1990.

[14] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Region-based convolutional networks for accurate object detection and segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 1, pp. 142–158, 2016. DOI: 10.1109/TPAMI.2015.2437384.

[15] H. Zhang, P. Zhang, X. Hu, *et al.*, *Glipv2: Unifying localization and vision-language understanding*, 2022. arXiv: 2206.05836 [cs.CV].

[16] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: An asr corpus based on public domain audio books," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015, pp. 5206–5210. DOI: 10.1109/ICASSP.2015.7178964.

[17] A. Ethics, *Acm code of ethics and professional conduct*, en-US, ACM Ethics - the Official Site of the Association for Computing Machinery's Committee on Professional Ethics, Jan. 2022. [Online]. Available: https://ethics.acm.org/.

[18] *IEEE Code of Ethics*, https://www.ieee.org/about/corporate/governance/p7-8.html.