Pathfinder

Helping you stay at home longer!

Table of contents

01 **Summary**

An overview of our project

02 **Solution and Concepts**

Why and how we chose our solution

03 **Decisions**

All the major decisions we made along the way

04 **Trials and Tribulations**

The lessons learned for our future work

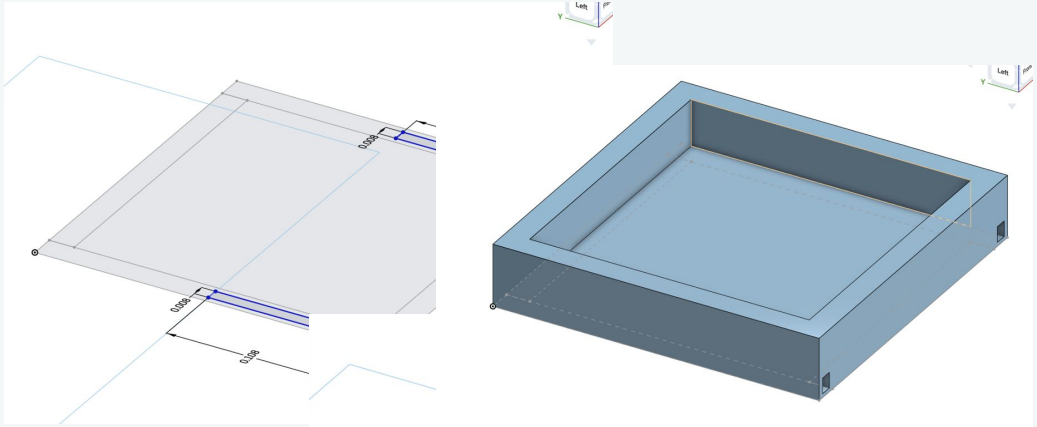
01

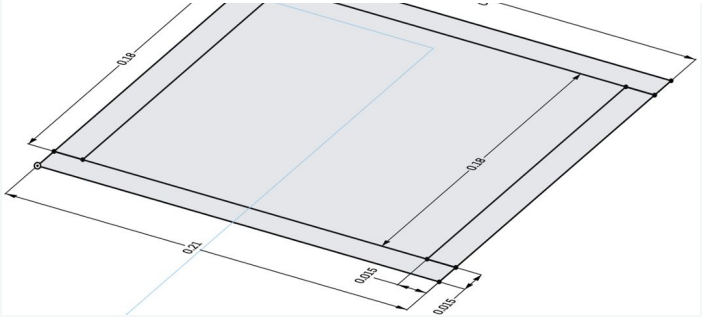
Summary

An overview of our project

Rory

Prototype I






```python
import cv2
import numpy as np
import time
from robomaster import robot

# Initialize RoboMaster
s1_robot = robot.Robot()
s1_robot.initialize(conn_type="sta")

# Start camera stream
s1_camera = s1_robot.camera
s1_camera.start_video_stream(display=True)

# Load YOLO Model
net = cv2.dnn.readNet("yolov3.weights", "yolov3.cfg")
layer_names = net.getUnconnectedOutLayersNames()
classes = open("coco.names").read().strip().split("\n")

print("RoboMaster initialized. YOLO loaded successfully.")

def detect_nearest_object():
    """ Detects the closest shape in front of the robot """
    frame = s1_camera.read_cv2_image(strategy="newest")

    # Convert image to grayscale
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

    # Apply edge detection
    edges = cv2.Canny(gray, 50, 150)

    # Find contours (shapes in the image)
    contours, _ = cv2.findContours(edges, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)

    if contours:
        # Identify the largest contour (assumed to be the closest object)
        largest_contour = max(contours, key=cv2.contourArea)

        # Get bounding box coordinates for the detected shape
        x, y, w, h = cv2.boundingRect(largest_contour)

        return (x, y, w, h)  # Return shape position

    return None  # No clear object detected

def follow_tracked_shape():
    """ Tracks and follows the nearest shape """
    while True:
        shape = detect_nearest_object()

        if shape:
            x, y, w, h = shape
            center_x = x + w // 2  # Get center of detected shape
            frame_center = 320  # Assuming a 640x480 resolution (center is 320px)

            if abs(center_x - frame_center) > 50:
                # If the object is off-center, adjust direction
                if center_x < frame_center:
                    print("Turning left...")
                    s1_robot.chassis.move(x=0, y=0, z=-15, xy_speed=30).wait_for_completed()
                else:
                    print("Turning right...")
                    s1_robot.chassis.move(x=0, y=0, z=15, xy_speed=30).wait_for_completed()
            else:
                print("Object centered. Moving forward...")
                s1_robot.chassis.move(x=0.3, y=0, z=0, xy_speed=0.5).wait_for_completed()
        else:
            print("Object lost. Waiting for confirmation...")
            time.sleep(2)

            # 🚀 INSERT VOICE COMMAND HERE 🚀
            # If no command to continue, stop tracking
            break
```

Prototype II

Finalized Print

Our stability tests confirmed that the platform can carry up to 10kg and is suitably big enough for the intended purposes.

Metrics

N°	Objective (Why)	Test Method (What)	Usage of Results (How)	Test Duration (When)	Type (What)	Objective (Why)	Fidelity	When to Realize
1	Validate voice command recognition accuracy	Issue various voice commands in different noise levels and accents, measuring success rate	Improve recognition reliability by adjusting sensitivity and filtering background noise	2 hours	Focused physical	Ensure real-world applicability of object detection	Medium	Before integrating full navigation system
2	Test response time of voice command execution	Measure delay between voice command input and robot's reaction	Optimize decision-making algorithm for faster execution	1.5 hours	Analytical	Improve real-time responsiveness of robot	Medium	Prior to testing full system integration

02

Solutions and Concepts

Why and how we chose our solution

Parker → Calum

Identified Needs

Lack of human judgement and understanding

AI lacks emotional intelligence and moral reasoning.

Detach from digital dehumanization

Human dignity must be preserved in digital interactions.

Impact on our relationship to technology

Technology should empower, not replace, human agency.

Proposed Solutions for Addressing Key Needs

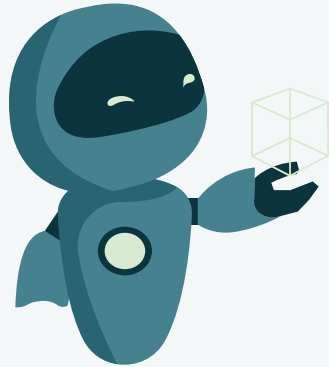
01 Home Assistance Robot

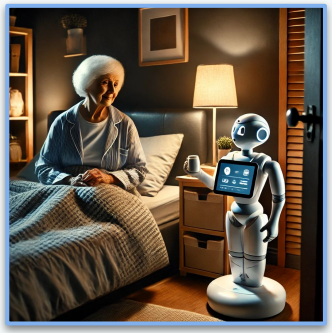
02 Police Assistant Robot

03 Delivery Robot

Solution	Pros	Cons
Robot Helping Police Officers	- Assists in dangerous or high-risk situations. - Could reduce human errors in some operations.	- May escalate the use of force for technology. - Lacks human understanding and reasoning - Does not diverge from war aspects
Home Assistant Robot (Elderly Help)	- Makes the robot seem like it is meant for peaceful means - Led by human interactions, keeping our decisions - Diverges from war aspects	- Lower Practicality - Privacy concerns about collecting data
Delivery Robot	- Empowers humans and serves mainly to make our lives easier - Diverges from War aspects	- Difficult to correctly implement - Dehumanizes humans into order numbers and is fully autonomous

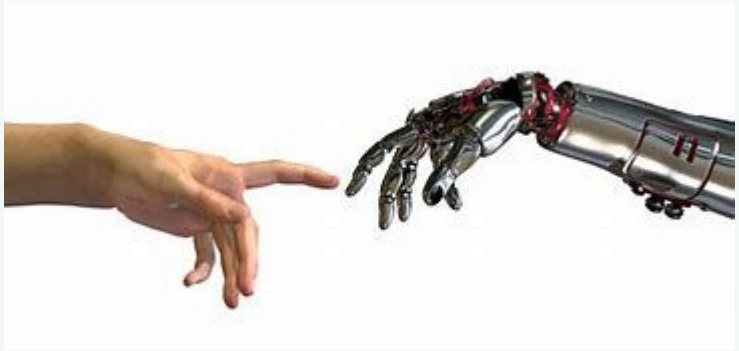
Chosen Concept:
Home Assistant Robot





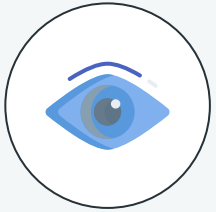
Why This Concept?

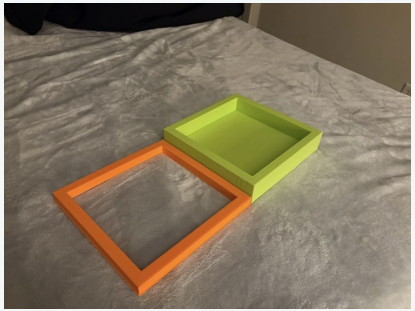
- Covers each identified client need of the highest priority:
  - Lack of human judgement and understanding
  - Digital dehumanization
  - Divergence from war aspects
- Provides useful real-world application



How is it Implemented?

- Using various different subsystems to ensure smooth function and navigation:
  - Object Recognition System
  - Navigation and Movement System
  - Decision Making System
- Medium-sized carrying platform for object transportation





03

Decisions

All the major decisions we made

Brayden

Major choices and design process

- How Robomaster's current functions best compliments our intended design
  - Manoeuvrability
  - Camera and swivel head
- Limitations in capability
  - Wheels
  - Battery
- Concept presentation and demonstration
  - Combine capability and client needs
- Modifications of robomaster
  - Attachments optimizations

Define systems and capabilities

Object recognition

Identifying and interpreting household objects

Voice recognition

Able to distinguish words and commands

Define systems and capabilities

Decision-making system

Ability to recognize and act on commands

Reaction time

Ensure fast and accurate execution of commands

04

Trials and Tribulations

The lessons learned for our future work

Bonita

Initial Challenges

Limited customization with RoboMaster S1

Object recognition struggled with lighting

Testing time constraints made iteration hard

Solutions Implemented

- Used 3D-printed parts
- Improved detection code
- Structured project timeline to improve time efficiency.

Technical Setbacks

Major Technical Issues

Object Recognition Accuracy: Poor performance under low-light conditions
Decision-Making Delays: System lagged when processing multiple commands
Platform Stability: Platform wobbled with heavier items

Fixes Applied

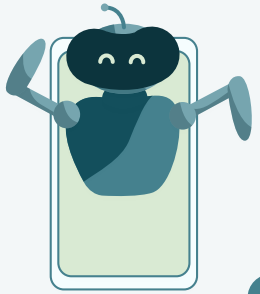
Refined detection logic
Added hybrid rule + AI system
Adjusted layout, added grip padding

User Feedback & Adjustments

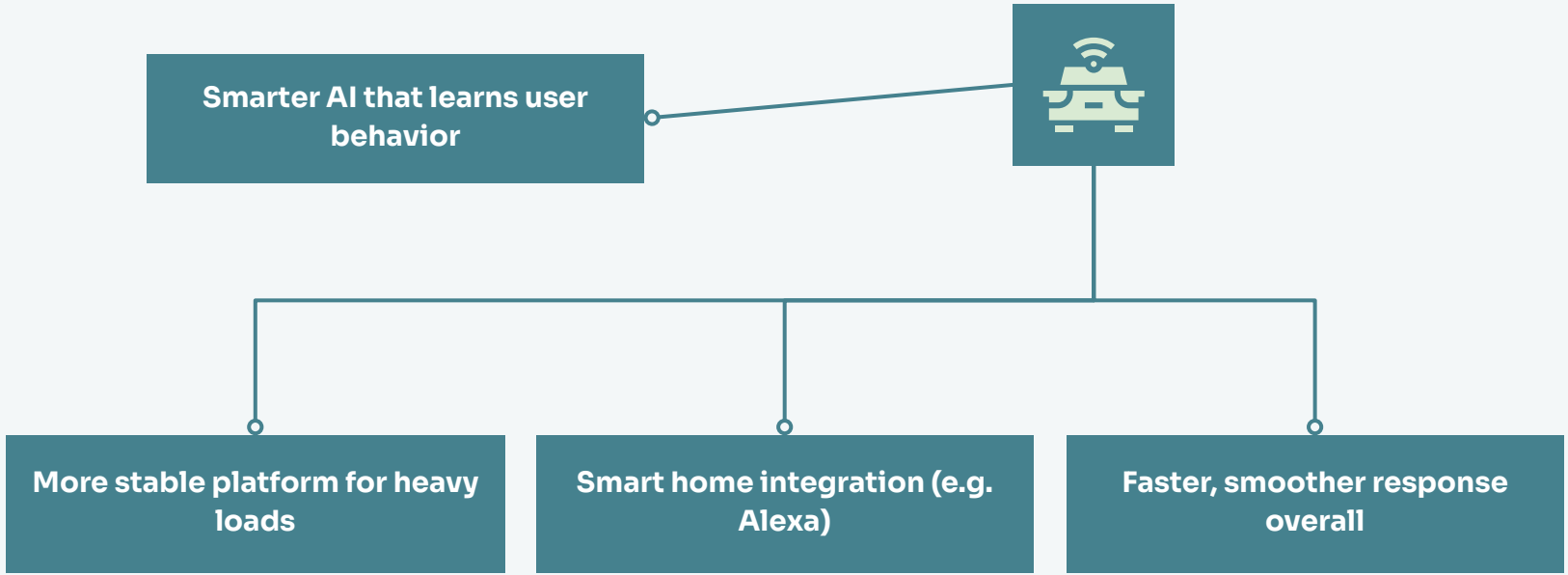
Tracking too slow → optimized response

Items felt unsafe → added latches + sensors

Battery drained fast → added power-saving mode



Future Enhancements



Questions?